

RH11-RS03

MAINT MODE DIAGNOSTIC
MD-11-DZRSE-C

EP-DZRSE-C-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

This microfiche card contains a grid of frames, each displaying diagnostic data for an MD-11 aircraft. The data is organized into columns and rows, with some frames containing headers such as 'TEST', 'PASS', 'FAIL', and 'TIME'. The frames are arranged in a regular grid pattern, with some frames appearing to be blank or containing minimal data. The overall layout is a dense grid of small, rectangular frames, each containing text and possibly some graphical elements like bar charts or tables. The data appears to be a series of test results or diagnostic codes, likely used for troubleshooting or maintenance purposes. The frames are arranged in a regular grid pattern, with some frames appearing to be blank or containing minimal data. The overall layout is a dense grid of small, rectangular frames, each containing text and possibly some graphical elements like bar charts or tables. The data appears to be a series of test results or diagnostic codes, likely used for troubleshooting or maintenance purposes.

MAINDEC-11-DZRSE-C RH11-RS03 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

95
96
97
98
99
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
149
150

1. ABSTRACT

THIS DIAGNOSTIC WILL LET THE OPERATOR SELECT ONE OF TWO MODES OF OPERATION. THE OPERATOR MAY SELECT WHICH DRIVE HE WANTS TESTED OR HE CAN LET THE PROGRAM SEQUENCE THROUGH ALL THE DRIVES ON THE SYSTEM.

THE FIRST PART OF THIS DIAGNOSTIC WILL TEST THE DRIVE REGISTERS ASSOCIATED WITH THE DRIVE UNDER TEST. THE PROGRAM WILL ALSO TEST THE RH CONTROLLER REGISTERS TO CONFIRM THAT, FOR THE MOST PART, THE CONTROLLER IS WORKING CORRECTLY.

THE SECOND PART OF THIS DIAGNOSTIC WILL TEST THE DRIVE IN "MAINTENANCE MODE".

THE RS03 HAS BEEN DESIGNED WITH BUILT-IN TEST CAPABILITIES. THIS "MAINTENANCE MODE" TEST CAPABILITY ISOLATES THE DIGITAL ELECTRONICS FROM THE ANALOG AND ALLOWS INDEPENDENT TESTING OF THE DIGITAL LOGIC. THEREFORE, FAILURES LOCATED ENTIRELY IN THE LOGIC CAN BE SEPARATED FROM FAILURES OCCURRING IN THE ANALOG ELECTRONICS OR THE HEAD/DISK SUBASSEMBLY.

NOTE

THIS DIAGNOSTIC WILL ALSO TEST RS03LA DRIVES. THEY MUST BE CONVERTED TO LOOK LIKE A RS03 TO BE TESTED. TURN DRIVES OFF LINE IF YOU DO NOT WISH TO TEST THEM.

1.1 DESIGN PHILOSOPHY

BY SETTING BIT 00 IN THE MAINTENANCE REGISTER, THE MAINTENANCE MODE LOGIC IS ENABLED, AND THE REMAINING READ/WRITE BITS IN THE MAINTENANCE REGISTER ARE SUBSTITUTED FOR THE CORRESPONDING SIGNALS NORMALLY ORIGINATING FROM THE HEAD/DISK SUBASSEMBLY. THE READ-ONLY BITS IN THE MAINTENANCE REGISTER REFLECT THE STATES OF MAJOR SIGNALS DURING DRIVE OPERATION. BY SETTING AND CLEARING THE READ/WRITE BITS IN PREDETERMINED SEQUENCES AND SIMULTANEOUSLY MONITORING THE READ-ONLY BITS, IT IS POSSIBLE TO VERIFY THE OPERATION OF ALL OF THE DRIVE'S LOGIC. THIS INCLUDES ALL DRIVE TIMING AS WELL AS THE LOGIC ASSOCIATED WITH READING AND WRITING DATA.

--CAUTION--

A THOROUGH UNDERSTANDING OF THE RS04 LOGIC IS REQUIRED TO UTILIZE THIS DIAGNOSTIC EFFECTIVELY. REFER TO SECTIONS 2 AND 3 OF THE "RS04 DECDISK SERVICE MANUAL" (DEC-00-HRS4A-A-D) FOR DESCRIPTIONS OF THE DRIVE LOGIC.

MAINDEC-11-DZRSE-C RH11-RS03 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

PAGE 4

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
205
206

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 WITH A MINIMUM OF 8K OF MEMORY AND AN RH11 CONTROLLER WITH A
RS03 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE SECTION 5 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESSES

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

STARTING ADDRESSES

1. STARTING ADDRESS 200

- A. SET SWITCHES (SEE SECTION 5), IF SWITCHLESS CPU, SIMPLY
- B. PRESS START
- C. THE PROGRAM WILL TYPE:
TEST ALL DRIVES? (Y OR N)
- D. IF THE OPERATOR TYPES "Y" THE PROGRAM WILL TEST ALL

FO1

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 6

207

RS03 DRIVES ON THE SYSTEM

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

E. IF THE OPERATOR TYPES "N" THE PROGRAM WILL TYPE
TYPE UNIT #

THE PROGRAM WILL ONLY TEST THAT DRIVE. THE PROGRAM
WILL THEN TYPE:

"ALL ERROR LIGHTS ON SELECTED UNIT SHOULD
BE ON - CHECK - THEN HIT CONT"

THE OPERATOR SHOULD CHECK THESE LIGHTS TO MAKE SURE
THAT THEY ARE ALL ON - THEN HIT CONTINUE. THE PROGRAM
WILL THEN START TESTING THE UNIT THAT WAS SELECTED.

2. STARTING ADDRESS 220

- A. SET SWITCHES (SEE SECTION 5), IF SWITCHLESS CPU SIMPLY
- B. PRESS START
- C. THE PROGRAM WILL THEN TEST ALL RS03 DRIVES ON THE SYSTEM.

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. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED
KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE
CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTRL 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 ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-R503 MAINTENANCE MODE DIAGNOSTIC

H01

MACY11 27(732) 25-SEP-76 10:44 PAGE 8

264

SWITCH SETTINGS ARE:

MAINDEC-11-DZRSE-C RH11-R503 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

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
318
319
320

SW<15> = 1 HALT ON ERROR
SW<14> = 1 LOOP ON TEST
SW<13> = 1 INHIBIT TYPEOUTS
SW<12> = 1 TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
SW<11> = 1 RUN MAINTENANCE MODE VERIFY TEST
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>

5.1 SUBROUTINE ABSTRACTS

THIS PROGRAM USES TRAP INSTRUCTIONS TO EXECUTE CLOCKING AND REGISTER CHECKING. THE TRAP INSTRUCTIONS THAT WE USED, ARE LISTED BELOW WITH A BREIF DESCRIPTION OF WHAT EACH ONE DOES.

5.1.1 CLRDK

TRAPS TO A TAG CALLED ".CLRDK". THIS ROUTINE CLEARS ALL REGISTERS BY SETTING THE "CLEAR BIT" IN RSCS2. (MOV#40,RHCS2) THE NUMBER OF THE UNIT UNDER TEST IS THEN RELOADED INTO RSCS2 AND THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE CLRDK INSTRUCTION.

5.1.2 MRDMD

TRAPS TO A TAG CALLED ".MRDMD". THIS ROUTINE PUTS THE DRIVE INTO MAINTENANCE MODE BY LOADING #000001 INTO RSMR AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRDMD INSTRUCTION.

5.1.3 MRINT

TRAPS TO A TAG CALLED ".MRINT". CLOCKS THE MAINTENANCE REGISTER TWICE WITH AN 11 AND A 1 AND RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRINT INSTRUCTION.

5.1.4 MRIND

TRAPS TO A TAG CALLED ".MRIND". CLOCKS AN INDEX PULSE INTO THE MAINTENANCE REGISTER THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRIND INSTRUCTION.

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RSG3 MAINTENANCE MODE DIAGNOSTIC

J01

MACY11 27(732) 25-SEP-76 10:44 PAGE 10

321
322

5.1.5 MRCLK

MAINDEC-11-DZRSE-C RH11-RS03 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

PAGE 7

TRAPS TO A TAG CALLED ".MRCLK". CLOCKS THE MAINTENANCE REGISTER WITH AN 11 AND A 1, UPDATES THE CLOCK COUNTER, AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MRCLK INSTRUCTION.

5.1.6 MRCK

TRAPS TO A TAG CALLED ".MRCK". THIS ROUTINE CHECKS THE MAINTENANCE REGISTER TO EQUAL THE VALUE FOLLOWING THE MRCK INSTRUCTION. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE CORRECT VALUE AND PRINTS OUT THE ERROR. IF THE MAINTENANCE REGISTER IS CORRECT, THE PROGRAM RETURNS TO THE INSTRUCTION FOLLOWING THE "HLT" INSTRUCTION.

5.1.7 DSCK

TRAPS TO A TAG CALLED ".DSCK". THIS ROUTINE CHECKS THE DRIVE STATUS REGISTER AND WORKS THE SAME WAY AS THE MRCK ROUTINE.

5.1.8 XBIT

TRAPS TO A TAG CALLED ".XBIT". THIS ROUTINE GETS ONE DATA BIT THAT IS CURRENTLY BEING WRITTEN FROM THE DATA BUFFER IN CORE AND STORES IT IN A LOCATION CALLED NOWOD. THE PREVIOUS CONTENTS OF NOWOD IS STORED IN LASTOD. THIS INFORMATION IS USED BY THE CLKD1 AND CLKD0 ROUTINES TO DETERMINE THE CORRECT STATE OF THE MWDB (BIT 12) BIT IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS04 WRITES 18 BIT WORDS) EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE XBIT INSTRUCTION.

5.1.9 CLKD1 AND CLKD0

TRAPS TO LOCATIONS ".CLKD1" AND ".CLKD0". THESE TWO ROUTINES USE THE DATA BITS RECEIVED FROM THE XBIT ROUTINE TO DETERMINE THE CORRECT STATE OF MWDB (BIT 12) IN RSMR WHEN WRITING. THESE ROUTINES ALSO CALCULATE THE CORRECT STATES OF THE CRCW, SB, AND LSR BITS IN RSMR AND DOES A COMPARE FOR THE CORRECT ANSWER. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE TRAP AND TYPES OUT THE ERROR. IF THE MAINTENANCE REGISTER WAS CORRECT, THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE "HLT."

5.1.10 RBIT

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
376
377
378

MAINDEC-11-DZRSE-C RH11-RS03 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

PAGE 8

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
434

TRAPS TO A TAG CALLED ".RBIT". THIS ROUTINE GETS THE ONE DATA BIT THAT ARE CURRENTLY BEING "READ" FROM THE DISK FROM THE INBUF DATA TABLE IN CORE AND STORES THAT BIT IN A LOCATION CALLED NOWOD. THE PROGRAM THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE RBIT INSTRUCTION.

5.1.11 CLKRI AND CLKRO

TRAPS TO LOCATIONS ".CLKRI" AND ".CLKRO". THESE TWO ROUTINES USING THE DATA BITS RECEIVED FROM THE RBIT ROUTINE SET AND CLEAR THE MRDB (BIT 2) BIT IN RSMR IN THE PROPER SEQUENCE CORRESPONDING TO THE DATA PATTERN WHICH IS BEING "READ". THESE ROUTINES ALSO CALCULATE THE CORRECT STATES OF THE CRCW AND SB BITS IN RSMR AND DOES A COMPARE FOR THE CORRECT ANSWER. IF THE MAINTENANCE REGISTER DOES NOT COMPARE, THE PROGRAM RETURNS TO THE "HLT" INSTRUCTION FOLLOWING THE TRAP AND TYPES OUT THE ERROR. IF THE MAINTENANCE REGISTER WAS CORRECT, THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE HLT.

5.1.12 MCLK1

TRAPS TO A TAG CALLED ".MCLK1". THIS ROUTINE CLOCKS THE MAINTENANCE REGISTER BY MOVING A 11 INTO RSMR. UPDATES THE CLOCK COUNTER AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLK1 INSTRUCTION.

5.1.13 MCLKO

TRAPS TO A TAG CALLED ".MCLKO". THIS ROUTINE CLOCKS THE MAINTENANCE REGISTER BY MOVING A 1 INTO RSMR. RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLKO INSTRUCTION.

5.1.14 MCLKB

TRAPS TO A TAG CALLED ".MCLKB". CLOCKS THE MAINTENANCE REGISTER WITH A 1 AND A 11, UPDATES THE CLOCK COUNTER, AND THEN RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE MCLKB INSTRUCTION.

5.1.15 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. THE CONTENTS OF LAD MAY BE USED

MO1

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 13

435

TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

MAINDEC-11-DZRSE-C RH11-RS03 BASIC FUNCTION DIAGNOSTIC
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

5.1.16 HLT

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

5.1.17 TRAPCATCHER

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

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

CS1,CS2,ER ETC. = RH11/RS04 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 OR AT 220

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

B02

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RCC3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 15

492

A BELL WILL RING WITHIN ONE AND A HALF MINUTES WITH ALL SWITCHES DOWN.

493
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

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

9. TEST DESCRIPTION

1. 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.

2. RESET TEST FOR REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSD8, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.

3. SET AND CLEAR ALL REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSD8 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.

4. TEST "CLEAR BIT" IN RSCS2

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

5. 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.

6. CLEAR DLT AND TRE

CLEAR BY SETTING TRE IN RSCS1 AND TEST.

7. LOAD RSD8 WITH ALL ONES AND ALL ZEROS

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

8. TEST FOR 66 LOCATIONS IN SILO

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

002

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 17

549

CHECKING THE OUTPUT FOR 66 WORDS.

550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605

9. TEST DLT ERROR

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

10. 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.

11. TEST PROGRAM INTERRUPT

THE PROGRAM FORCES A INTERRUPT BY MOVING A 300 INTO RSCS1.

12. MAINTENANCE TIMING TEST

THE FOLLOWING TEST ON THE RS03 DISK IS A SINGLE-STEPPED MAINTENANCE MODE TEST ON THE RS03 TIMING LOGIC. THE ACTUAL DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE REGISTER, I.E. THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE TIMING LOGIC. WE ARE TESTING THE ENTIRE "TIMING TRACK", INDEX PULSE FUNCTION, RESYNC AREA, SECTOR COUNTER, ETC.

- PUT DRIVE INTO MAINTENANCE MODE.
- ASSERT INDEX PULSE TO INITIALIZE DRIVE TIMING LOGIC.
- INDEX PULSE SHOULD CLEAR LOOK-AHEAD REGISTER.
- CLOCK TIMING TO STEP THROUGH RESYNC PERIOD.
- CHECK FOR SECTOR PULSE.
- PERFORM MAINTENANCE CLOCK OPERATION TO CHECK FOR 64 SECTOR PULSES.
- THE LOOK-AHEAD REGISTER SHOULD NOW POINT TO THE CURRENT SECTOR.
- REPEAT STEPS TO CLOCK THROUGH ALL THE SECTORS TO CHECK SECTOR COUNT.

13. SECTOR FRACTION TEST

CLOCK THROUGH AN ENTIRE TRACK IN MAINTENANCE MODE WHILE CHECKING FOR THE PROPER OPERATION OF THE LOOK-AHEAD REGISTER AND THE SECTOR FRACTION COUNTER.

- INITIALIZE DRIVE AND STEP THROUGH RESYNC AREA.
- CHECK FOR SECTOR PULSE.
- LOOK-AHEAD REGISTER SHOULD = 0.
- STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA AREA WHILE CHECKING THE SECTOR FRACTION.
- CHECK FRACTIONS TO CHANGE AFTER THE CORRECT NUMBER OF MAINTENANCE CLOCKS.

F02

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 19

606
607

WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND FRACTION
IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK --

608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663

HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL CLEAR THE FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

14. DISK ILLEGAL FUNCTION TEST

TEST ILLEGAL FUNCTION (ILF) IN RSER. SEND AN ILLEGAL FUNCTION CODE TO THE DRIVE CONTROL REGISTER WITHOUT SETTING THE GO BIT. THE "ILF" BIT SHOULD NOT BE SET. THE "GO" BIT IS THEN SET. A CHECK IS THEN MADE FOR "ATA" AND "ERR" TO BE SET IN THE DRIVE STATUS REGISTER (RSDS) AND "ILF" IN THE DRIVE ERROR REGISTER (RSER). ALL ILLEGAL FUNCTION CODES ARE CHECKED.

15. TEST THE DRIVE NO-OP CODES 1 AND 21

THIS IS TESTED WITH AND WITHOUT ERRORS BEING SET TO PROVE THAT IT DOESN'T CHANGE ANYTHING.

16. DRIVE SEARCH TEST 1

A DRIVE SEARCH FUNCTION IS GIVEN TO THE DRIVE FOR SECTOR 3. (SECTOR 41 IF SECTOR INTERLEAVING IS ENABLED) THE POSITIONING IN PROGRESS BIT (PIP) AND THE DRIVE READY BIT (DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.

17. DRIVE SEARCH TEST 2

THIS TEST INITIALIZES A DRIVE SEARCH FUNCTION FOR SECTOR 0 WHEN THE DRIVE IS CURRENTLY AT THE DESIRED SECTOR, THE SEARCH FUNCTION SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION AND REACHES THE BEGINNING OF THE DESIRED SECTOR.

18. REGISTER MODIFICATION REFUSED TEST

RMR IN THE DRIVE ERROR REGISTER (RSER) SHOULD SET BY TRYING TO MODIFY ONE OF THREE DRIVE REGISTERS WHILE THE DRIVE IS BUSY DURING A DRIVE SEARCH FUNCTION.

1. RSCS1
2. RSDA
3. RSER

TEST THAT RMR DOES NOT SET WHEN MODIFYING THE ATTENTION SUMMARY REGISTER (RSAS).

19. DRIVE SELECT TEST

THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES. THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC

H02

MACY11 27(732) 25-SEP-76 10:44 PAGE 21

664
665

TRIES TO LOAD ITS REGISTER WITH ALTERNATE ONES AND ZEROS.
THIS SHOULD CAUSE "NED" TO SET IN RSCS2. THE PROGRAM

666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721

RE-SELECTS THE DRIVE UNDER TEST AND CHECKS ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.

20. MAINTENANCE WRITE TEST

THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR WRITE TEST. WE ARE TESTING THE COMPLETE DATA PATH FOR A DATA TRANSFER TO THE DISK. MILLER ENCODED DATA TO BOTH SURFACES IS CHECKED ALONG WITH CORRECT GENERATION OF THE CRC WORD AT THE END OF THE SECTOR. INDEX PULSES, RESYNC, TIMING PREAMBLE, AND SECTOR PULSES ARE ALSO CHECKED.

21. MAINTENANCE READ TEST

THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS NOT TESTED IN MAINTENANCE MODE.)

22. MAINTENANCE MODE DATA WRITE CHECK TEST

A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION. WITHIN THE RS04, A WRITE CHECK FUNCTION IS IDENTICAL TO A READ FUNCTION.

23. MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)

THE RS03 DISK IS SET UP TO READ (IN MAINTENANCE MODE) ONE SECTOR OF A SPECIALLY CREATED DATA PATTERN WHICH LEAVE ONLY ONE BIT SET IN THE CRC REGISTER PRIOR TO CHECKING THE CRC WORD. THE CORRESPONDING CRC WORD IS THEN "READ", RESULTING IN NO DCK ERROR. THE DATA PATTERN IS THEN MODIFIED (BY SHIFTING) AND THE ENTIRE READ SEQUENCE REPEATED UNTIL ALL 16 BITS IN THE CRC REGISTER HAVE BEEN CHECKED.

24. MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ". THIS CAUSES A DCK ERROR. THE READ SEQUENCE IS REPEATED 16 TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A DCK ERROR.

25. IGNORE FUNCTION TEST

PUT THE DISK IN MAINTENANCE MODE AND SET ERROR CONDITIONS IN THE DRIVE ERROR REGISTER (RSER). TRY TO START A READ TRANSFER. THE "GO" BIT IN RSCS1 SHOULD NOT SET. MISSED TRANSFER ERROR (MXF) SHOULD SET IN RSCS2 WHICH IN TURN SHOULD CAUSE "TRE" AND "SC" TO SET IN RSCS1.

J02

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-REQ3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 23

722

26. INVALID ADDRESS TEST



723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
47
3
19
50
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778

FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE CONTROL REGISTER (RSCS1).

27. DISK OPERATION INCOMPLETE (OPI) ERROR TEST

PUT DISK IN MAINTENANCE MODE AND START A READ COMMAND. THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE ROTATION OF THIS DISK SURFACE. THE THIRD INDEX PULSE SHOULD CAUSE OPERATION INCOMPLETE (OPI) TO SET IN THE DRIVE ERROR REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSCS).

28. PARITY ERROR TEST

SET "PAT" BIT IN RSCS2. WRITE A DRIVE REGISTER. "PAR" SHOULD SET IN THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS AND 'SC' TO SET IN RSCS1.

29. MAINTENANCE MODE INTERRUPT TEST

IN THIS TEST THE INTERRUPT ENABLE (I.E.) BIT IS SET. A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMR" ERROR IS THEN CAUSED WHILE THE FIRST SECTOR IS BEING WRITTEN. WHEN THE FUNCTION IS COMPLETED, THE DRIVE SHOULD INTERRUPT.

30. DISK ADDRESS OVERFLOW (AOE) TEST

SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77 TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER (LBT) BIT TO SET IN THE RSDS REGISTER.

31. MAINTENANCE VERIFY TEST

THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH REGISTER" FOR IT WILL ACTUALLY WRITE DATA ONTO THE DISK. IT WILL WRITE ONE TRACK OF ALL ONES. THE DRIVE IS THEN PLACED IN MAINTENANCE MODE AND IT WILL THEN WRITE ONE SECTOR OF THE SAME TRACK WITH ALL ZEROS. THE DRIVE IS THEN TAKEN OUT OF "MAINTENANCE MODE" AND THE TRACK IS THEN READ. THE TRACK SHOULD CONTAIN ALL ONES.

				SWITCH	USE
				-----	-----
779					
780					
781					
782		100000		SW15= 100000	; HALT ON ERROR
783		040000		SW14= 40000	; LOOP ON TEST
784		020000		SW13= 20000	; INHIBIT ERROR TYPEOUTS
785		010000		SW12= 10000	; TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
786		004000		SW11= 4000	; RUN MAINTENANCE MODE VERIFY TEST
787		002000		SW10= 2000	; 0 - BELL ON PASS COMPLETE
788					; 1 - BELL ON ERROR
789		001000		SW9= 1000	; LOOP ON ERROR
790		000400		SW8= 400	; LOOP ON TEST IN SW<7:0>
791		000000			; TRAP CATCHER FROM 0 - 776
792		000046			; HOOKS FOR ACT 11
793	000046	021736			
794		000052			
795	000052	040000			
796					
797		000174			
798	000174	000000			
799	000176	000000			
800					
801		000200			
802	000200	000137	000232		
803					
804		000220			
805	000220	052737	000100 001146		
806	000226	000137	001240		
807					
808	000232	042737	000100 001146		
809	000240	000772			
810					
811					

812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844

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

GOOD=
BAD=

N= 1
HLT= EMT
PS= 177776
PSW= PS
BELL= 7
R0= %0
R1= %1
R2= %2
R3= %3
R4= %4
R5= %5
SP= %6
PC= %7
BIT0= 1
BIT1= 2
BIT2= 4
BIT3= 10
BIT4= 20
BIT5= 40
BIT6= 100
BIT7= 200
BIT8= 400
BIT9= 1000
BIT10= 2000
BIT11= 4000
BIT12= 10000
BIT13= 20000
BIT14= 40000
BIT15= 100000
%1
%0

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

;FOR GOOD DATA
;FOR BAD DATA

845 001000
 846 001000
 847 001000 000000
 848 001002 000000
 849 001004 000000 000000
 850 001010 000000
 851 001012 000000
 852 001014 001000
 853 001016 177564
 854 001020 177560
 855 001022 177562
 856 001024 177566
 857 001026 177570
 858 001030 177570
 859
 860 001100
 861
 862
 863
 864 001100 172040
 865 001102 172050
 866 001104 172042
 867 001106 172044
 868 001110 172046
 869 001112 172052
 870 001114 172054
 871 001116 172056
 872 001120 172060
 873 001122 172062
 874 001124 172064
 875 001126 172066
 876 001130 000204
 877 001132 000206
 878 001134 172041
 879 001136 172051
 880 001140 172043
 881 001142 172045
 882

. = 1000

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

. = 1100

; DISK I/O REGISTERS

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

; DISK CONTROL + STATUS REGISTER
 ; DISK CONTROL + STATUS REGISTER
 ; WORD COUNT REGISTER
 ; BUS ADDRESS
 ; DISK ADDRESS (DESIRED ADDRESS)
 ; DRIVE STATUS
 ; ERROR REG.
 ; ATTENTION SUMMARY
 ; LOOK AHEAD
 ; DATA BUFFER REGISTER
 ; MAINTENANCE REGISTER
 ; DRIVE TYPE REGISTER
 ; INTERRUPT VECTOR
 ; INTERRUPT PRIO. VECTOR
 ; ODD BYTE ADD FOR CS1
 ; ODD BYTE ADD FOR CS2
 ; ODD BYTE ADD FOR CW
 ; ODD BYTE ADD FOR BA

883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924

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

925
926
927 001144 000000
928 001146 000000
929 001150 000000
930 001152 000000
931 001154 000000
932 001156 000000
933 001160 000000
934 001162 000000
935 001164 000000
936 001166 000000
937 001170 000000
938 001172 000000
939 172100
940 001174 000000
941 001176 000000
942 001202 000000
943 001204 000000
944 001206 000000
945 001210 000000
946 001212 000000
947 001214 000000
948 001216 000000
949 001220 000000
950 001222 000000
951 001224 000000
952 001226 000000
953 001230 000000
954 001232 000000
955 001234 000000
956 001236 000000

000000

:WORKING LOCATIONS

FLAG2: 0
FLAG3: 0
LSTEV: 0
LSTOD: 0
NOWEV: 0
NOWOD: 0
RSD: 0
UNNUM: 0
UNITSV: 0
UNCMP: 0
ONCEE: 0
TIMSV: 0
MPRO=172100
SAVEE: 0
MCCNT: 0,0
WCRC: 0
REPT: 0
REPT1: 0
CLKCNT: 0
INBIT: 0
WK15: 0
WORK: 0
WORK0: 0
WORK1: 0
WORK2: 0
WORK3: 0
WORK4: 0
WORK5: 0
WORK6: 0
LAFLAG: 0

:SECOND FLAG WORD
:3RD FLAG WD
:LAST EVEN BIT TRANSFERED
:LAST ODD BIT TRANSFERED
:PRESENT EVEN BIT BEING XFERED
:PRESENT ODD BIT BEING XFERED
:SAME
:UNIT CURRENTLY BEING TESTED
:SET BIT=UNIT ON BUS
:FOR COMPARING FOR # OF DEVICE
:DID WE TEST ANY DRIVES
:SAVE LOC FOR TIME
:PARITY REG
:WORK LOC
:MAINT CLOCK COUNT
:WORK LOC FOR CREATING CRC WORD
:REPEAT COUNTER
:REPEAT COUNTER
:REPEAT COUNTER
:CLOCK COUNTER FOR EACH WORD
:USED IN CRC CAL ROUTINE
:USED IN CRC CAL ROUTINE

:FLAG FOR LA DONE TYPE OUT

```
957 ;DISCRIPTION OF BITS IN LOCATION ONCEE
958
959 ;BIT0 MEANS FOUND DRIVE
960 ;BIT1 ERROR DO NOT CHANGE ILLEGAL FUNCTION
961 ;BIT2 ERROR FLAG
962 ;BIT3 TESTING CODE 21 FLAG
963 ;BIT5 TYPEOUT CLOCK COUNT
964 ;BIT6 1ST TRANSFER WORD FLAG
965 ;BIT7 WRITTING LAST WORD OF SECOTR
966 ;BIT8 TRANSFERRING CRC WORD
967 ;BIT9 FOR INTERLEAVED DRIVES
968 ;BIT10 1ST TIME FLAG IN SECTOR FRACTION TEST
969 ;BIT11 DO TKSEL TEST
970 ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
971 ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
972 ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
973 ;BIT15 MEANS ERROR FOUND
974
975 ;DISCRIPTION OF BITS IN LOCATION FLAG2
976
977 ;BIT0 SWITCH FOR RWCLK IN MR REG
978 ;BIT1 MAINTENANCE MODE VERIFY TEST
979 ;BIT2 IN WRITE CK TEST FOR CLKRI ROUTINE
980 ;BIT3 DONE 1ST CRC WD IN CRC TEST
981 ;BIT4 1ST TIME THROUGH IN CRC TEST
982 ;BIT5 IN CRC TEST
983 ;BIT7 DOING FIRST XFER WD IN XBIT
984 ;BIT8 XFER DATA BITS 16 AND 17 IN XBIT ROUTINE
985 ;BIT9 SAME
986 ;BIT10 XFER CRC BITS 16 AND 17 IN XBIT ROUTINE
987 ;BIT11 USED IN RBIT ROUTINE FOR DATA BITS 17 AND 16
```

E03

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 31

```

988 001240 012706 000500          BEGIN:  MOV    #500,SP          ;SET STACK TO *** 500 ***
989 001244 012737 025760 000024  MOV    #.POWER,2#24      ;SET UP PF VECTOR
990 001252 012737 000340 000026  MOV    #340,2#26         ;LOCK OUT THE WORLD
991 001260 012737 025406 000030  MOV    #.HLT,2#30        ;SET EMT VECTOR
992 001266 012737 000340 000032  MOV    #340,2#32         ;LOCK UP
993 001274 012737 026414 000034  MOV    #.TRAP,2#34       ;SET TRAP VECTOR
994 001302 012737 000340 000036  MOV    #340,2#36         ;LOCK UP
995 001310 005037 001000          CLR    ICNT              ;INIT ICNT
996 001314 005037 001010          CLR    LAD               ;INIT LAD
997 001320 042737 000020 001146  BIC    #BIT4,FLAG3       ;CLEAR TEST ONLY ONE DRIVE FLAG
998 001326 005037 001236          CLR    LAFLAG            ;CLEAR TYPE FLAG FOR LA DSK
999 001332 042737 177677 001144  BIC    #177677,FLAG2
1000 001340 042737 153777 001170  BIC    #153777,ONCEE
1001 001346 032737 000100 001146  BIT    #BIT6,FLAG3       ;TEST ALL DRIVES?
1002 001354 001402          BEQ    5$                ;ASK
1003 001356 000137 002006          JMP    2#MULTII
1004 001362          5$:
1005 001362 104402 001366          TYPE    ,.+2             ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
1006 001424 104412          RDLIN
1007 001426 122737 000131 026374  CMPB   #'Y,INPUT         ;TEST FOR YES
1008 001434 001564          BEQ    MULTII            ;YES
1009 001436 052737 000020 001146  BIS    #BIT4,FLAG3       ;SET TEST ONLY ONE DRIVE FLAG
1010 001444          1$:
1011 001444 104402 001450          TYPE    ,.+2             ;.ASCIZ "TYPE UNIT #"
1012 001464 104410          RDOCT
1013 001466 012604          MOV    (6)+,R4           ;GET NUMBER
1014 001470 022704 000010          CMP    #10,R4           ;CORRECT #
1015 001474 101763          BLOS   1$                ;NO
1016 001476 010437 001162          MOV    R4,UNNUM         ;SET UNIT #
1017 001502 005002          CLR    R2               ;CLEAR WORK AREA
1018 001504 000261          SEC
1019 001506 006102          2$:  ROL    R2             ;SET CARRY
1020 001510 005704          TST   R4                ;SET WORK BIT
1021 001512 001402          BEG    3$                ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
1022 001514 005304          DEC   R4                ;YES
1023 001516 000773          BR    2$                ;NO TRY AGAIN
1024 001520 010237 001164          3$:  MOV    R2,UNITSV        ;TEST AGAIN
1025 001524 010237 001166          MOV    R2,UNCOMP        ;SET DRIVE BIT IN UNITSV
1026 001530 013777 001162 177344  MOV    UNNUM,2#RSCS2     ;SET UNIT COMPARE
1027 001536 012777 177777 177350  MOV    #-1,2#RSER       ;LOAD DRIVE
1028 001544 022777 000004 177354  CMP    #4,2#RSOT        ;LOAD ERRORS
1029 001552 001033          BNE   6$                ;RS04LA?
1030 001554 104402 001560          6$:  TYPE    ,.+2             ;.ASCIZ <15><12>"THIS IS A RS03LA IT WILL BE TESTED AS A
1031 001642          6$:
1032 001642 104402 001646          TYPE    ,.+2             ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
1033 001756 000000          HALT
1034 001760 023777 001164 177130  CMP    UNITSV,2#RSAS    ;WAIT FOR LIGHTS TO BE CHECKED
1035 001766 001405          BEQ    4$                ;DID CORRECT ATA SET
1036 001770 017700 177122          MOV    2#RSAS,BAD       ;GET RSAS
1037 001774 013701 001164          MOV    U.ITSV,GOOD      ;GET CORRECT AND
1038 002000 104000          HLT
1039
1040
1041 002002 000137 002564          4$:  JMP    NOWGO            ;RSAS=BAD GOOD=CORRECTIONS
                                     ;ATA BIT SHOULD SET FOR ERRORS
                                     ;WERE SET IN RSER
                                     ;START TESTING

```

```

1042                                     ;NOW TEST FOR DRIVES
1043
1044 002006 012701 000010          MULTII: MOV      #8, R1          ;PUT 8 INTO R1 FOR COUNT
1045 002012 005077 177064          CLR      @RSCS2        ;SET DEVICE TO ZERO
1046 002016 012777 177777 177070 TRY:  MOV      #-1, @RSER      ;CAUSE AN ERROR +SETS BIT IN RSAS REG
1047 002024 005301                   DEC      R1            ;DO A MAXIMUM OF 8 TIMES
1048 002026 001403                   BEQ     DVNUM          ;TESTED FOR ALL DRIVES GET OUT
1049 002030 005277 177046          INC     @RSCS2        ;INCREMENT DRIVE UNIT
1050 002034 000770                   BR     TRY            ;REPEAT FOR NEXT DRIVE
1051 002036 017737 177054 001164 DVNUM: MOV     @RSAS, UNITSV    ;SAVE
1052 002044 012737 000401 001166   MOV     #401, UNCMP   ;SETUP TO CMP WITH UNITSV
1053 002052 012737 000000 001162   MOV     #0, UNNUM     ;PUT 0 INTO UNIT NO.
1054 002060 032777 020000 176740   BIT    #BIT13, @SWR   ;INHIBIT TYPE OUT?
1055 002066 001015                   BNE    STTEST        ;YES
1056 002070 104402 002074          TYPE    ,.+2         ;.ASCIZ <15><12>"TESTING UNIT "
1057 002114 042737 100000 001170   BIC    #BIT15, ONCEE  ;CLEAR ERROR FLAG
1058 002122 033737 001166 001164 STTEST: BIT    UNCMP, UNITSV    ;IS THIS DRIVE ON THE SYSTEM
1059 002130 001510                   BEQ    TRYNX         ;NO
1060 002132 013777 001162 176742   MOV    UNNUM, @RSCS2 ;YES PUT UNIT # INTO CS2
1061 002140 022777 000000 176760 3$:  CMP     #0, @RSDT     ;IS THIS A RS03?
1062 002146 001454                   BEQ    1$            ;YES
1063 002150 022777 000001 176750   CMP     #1, @RSDT     ;IS IT A RS03?
1064 002156 001450                   BEQ    1$            ;YES
1065 002160 022777 000004 176740   CMP     #4, @RSDT     ;RS03LA?
1066 002166 001071                   BNE    TRYNX         ;NO
1067 002170 005737 001236          TST    LAFLAG        ;1ST TIME
1068 002174 001041                   BNE    1$            ;NO
1069 002176 032777 020000 176622   BIT    #BIT13, @SWR   ;INHIBIT TYPEOUT
1070 002204 001061                   BNE    4$            ;YES
1071 002206 104402 002212          TYPE    ,.+2         ;.ASCIZ <15><12>
1072 002216 013746 001162          MOV    UNNUM, -(6)    ;PUT UNNUM ON STACK
1073 002222 104406                   TYPES   ;TYPE STACK IN OCTAL - SUPRESS
1074 002224 104402 002230          TYPE    ,.+2         ;.ASCIZ " IS A RS03LA WILL TEST IT LIKE A RS03"<15><12>
1075 002300 032777 020000 176520 1$:  BIT    #BIT13, @SWR   ;INHIBIT TYPE OUT?
1076 002306 001020                   BNE    4$            ;YES
1077 002310 032737 100000 001170   BIT    #BIT15, ONCEE ;ANY ERRORS?
1078 002316 001404                   BEQ    5$            ;NO
1079 002320 104402 002324          TYPE    ,.+2         ;.ASCIZ <15><12><12>
1080 002330                   5$:
1081 002330 013746 001162          MOV    UNNUM, -(6)    ;PUT UNNUM ON STACK
1082 002334 104406                   TYPES   ;TYPE STACK IN OCTAL - SUPRESS
1083 002336 104402 000040          TYPE    ,40         ;TYPE SPACE
1084 002342 042737 100000 001170   BIC    #BIT15, ONCEE ;CLEAR ERROR FLAG
1085 002350 000505                   BR     NOWGO         ;NOW TEST
1086 002352 032737 000020 001146 TRYNX: BIT    #BIT4, FLAG3    ;MULTI DRIVE
1087 002360 001074                   BNE    DONEE        ;NO
1088 002362 006337 001166          ASL    UNCMP         ;CHECK NEXT BIT FOR DRIVE
1089 002366 103403                   BCS    CHCKDV       ;DID WE TEST ANY REG?
1090 002370 005237 001162          INC    UNNUM         ;INC UNIT #
1091 002374 000652                   BR     STTEST        ;CHECK FOR NEXT DRIVE

```



```

1092 002376 032737 000001 001170 CHCKDV: BIT      #BIT0,ONCEE      ;DID WE TEST ANY DRIVES?
1093 002404 001062          BNE      DONEE        ;YES WE DID TEST A DRIVE
1094 002406 012737 100000 001166 MOV      #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
1095 002414 005037 001162          CLR      UNNUM        ;ANY AS BITS, THUS DEFAULTS TO
1096 002420 032777 020000 176400 BIT      #BIT13,ASWR   ;INHIBIT TYPE OUT?
1097 002426 001050          BNE      45           ;YES
1098 002430 013746 001162          MOV      UNNUM,-(6)   ;PUT UNNUM ON STACK
1099 002434 104406          TYPES          ;TYPE STACK IN OCTAL - SUPRESS
1100 002436 104402 000040          TYPE      ,40        ;TYPE SPACE
1101 002442 104402 002446          TYPE      ,+2        ;ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
1102 002540 012737 000001 001166 MOV      #1,UNCMP     ;SETUP TO TEST UNIT 0
1103 002546 000000          HALT                    ;WAIT
1104 002550 000405          45: BR      NOWGO     ;TEST DRIVE 0
1105 002552 012737 177777 001236 DONEE: MOV      #-1,LAFLAG ;SET LA FLAG DRIVE IDENTIFICATION
1106 002560 000137 021672          JMP      DONEE        ;GET OUT

;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
;TO CLEAR ALL THE RH AND RS REGISTERS

1111 002564 052737 000001 001170 NOWGO: BIS      #BIT0,ONCEE ;SET FOUND DRIVE FLAG
1112 002572 013737 025404 001172 MOV      TIMES,TIMSV ;SAVE TIME
1113 002600 012737 000001 025404 MOV      #1,TIMES    ;ONLY TEST ONCE
1114 *****
1115 ;TEST 1 RESET TEST FOR REGISTERS
1116 *****
1117 TEST1: SCOPE
1118 002606 104400          MOVL     #340,ASPS    ;LOCK OUT INTERRUPTS
1119 002610 012737 000340 177776 MOV      UNNUM,ARSCS2 ;LOAD UNIT NO.
1120 002616 013777 001162 176256 MOV      #177776,ARSCS1 ;SET ALL
1121 002624 012777 177776 176246 MOV      #177777,ARSCS1 ;POSSIBLE R/W
1122 002632 012777 177777 176246 MOV      #177777,ARSCS1 ;BITS IN THESE REGISTERS
1123 002640 012777 177777 176242 MOV      #177777,ARSDA
1124 002646 012777 177777 176240 MOV      #177777,ARSER
1125 002654 012777 177777 176242 MOV      #177777,ARSMR
1126 002662 012777 177777 176214 MOV      #177777,ARSWC
1127 002670 012777 177737 176204 MOV      #177737,ARSCS2
1128          RESET      ;CLEAR ALL BITS IN ALL REG.
1129
1130 ;TEST RSCS2 FOR CLEARED BITS
1131 002700 022777 000100 176174 CMP      #100,ARSCS2 ;DID THESE BITS GET CLEARED?
1132 002706 001401          BEQ      +4           ;YES
1133 002710 104200          HLT      !CS2        ;(417) SHOULD BE CLEARED IN CS2
1134 002712 013777 001162 176162 MOV      UNNUM,ARSCS2 ;PUT # OF UNIT IN TEST IN CS2
1135 002720 022777 010600 176164 CMP      #10600,ARSDS ;IS DPR AND MOL SET?
1136 002726 001401          BEQ      +4           ;YES
1137 002730 104040          HLT      !DS         ;NO WHY NOT?
1138
1139 ;TEST CONTROL AND STATUS REG 1
1140 002732 022777 004200 176140 CMP      #4200,ARSCS1 ;DID THE READY BIT SET?
1141 002740 001401          BEQ      +4           ;YES
1142 002742 104001          HLT      !CS1        ;READY SHOULD BE SET

```

H03

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST1

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC
RESET TEST FOR REGISTERS

MACY11 27(732) 25-SEP-76 10:44 PAGE 34

```

1143           ;TEST BUS ADDRESS REGISTER
1144
1145 002744 005777 176136           TST   @RSBA           ;IS BA REG. CLEARED
1146 002750 001401                   BEQ   :+4           ;YES
1147 002752 104020                   HLT   !BA           ;SHOULD BE 0
1148
1149           ;TEST DISK ADDRESS REGISTER
1150
1151 002754 005777 176130           TST   @RSDA           ;IS DA CLEARED
1152 002760 001401                   BEQ   :+4           ;YES
1153 002762 104004                   HLT   !DA           ;SHOULD BE 0
1154
1155           ;TEST ERROR REG RSER
1156
1157 002764 005777 176124           TST   @RSER           ;DID RSER CLEAR?
1158 002770 001401                   BEQ   :+4           ;YES
1159 002772 104002                   HLT   !ER           ;BITS(157015) SHOULD BE CLEARED
1160
1161           ;TEST RS MAINTENANCE REGISTER
1162
1163 002774 032777 000077 176122     BIT   #77,@RSMR       ;DID THESE BITS GET CLEARED
1164 003002 001401                   BEQ   :+4           ;YES
1165 003004 104220                   HLT   !MR           ;BITS(77) SHOULD BE 0
1166
1167           ;TEST WC REG IT SHOULD NOT CHANGE
1168
1169 003006 022777 177777 176070     CMP   #177777,@RSWC   ;DID IT CHANGE?
1170 003014 001401                   BEQ   :+4           ;NO
1171 003016 104010                   HLT   !WC           ;RESET SHOULD NOT MODIFY RSWC
1172
1173           ;TEST RSAS
1174
1175 003020 005777 176072           TST   @RSAS           ;IS REG CLEAR
1176 003024 001401                   BEQ   :+4           ;YES
1177 003026 104100                   HLT   !AS           ;NO

```

```

1178
1179
1180
1181 003030 104400
1182
1183 003032 012737 000340 177776 TTAGG: MOV #340, @#PS ; LOCK OUT INTERRUPTS
1184 003040 013777 001162 176034 MOV UNNUM, @RSCS2 ; LOAD UNIT NO.
1185 003046 012777 043576 176024 MOV #43576, @RSCS1 ; SET ALL
1186 003054 012777 020417 176020 MOV #20417, @RSCS2 ; ALL
1187 003062 012777 177777 176016 MOV #177777, @RSBA ; POSSIBLE
1188 003070 012777 177777 176012 MOV #177777, @RSDA ; REGISTERS
1189 003076 012777 177017 176010 MOV #177017, @RSER
1190 003104 012777 177777 176010 MOV #177777, @RSDB
1191 003112 012777 177777 175764 MOV #177777, @RSWC
1192 003120 012777 020417 175754 MOV #20417, @RSCS2
1193 003126 012777 000071 175770 MOV #71, @RSMR
1194 003134 012777 000040 175740 MOV #40, @RSCS2 ; CLEAR ALL BITS
1195 003142 022777 000100 175732 CMP #100, @RSCS2 ; DID THE RIGHT BITS CLEAR?
1196 003150 001401 BEQ +4 ; YES
1197 003152 104200 HLT ;CS2 ; (417) SHOULD BE CLEARED IN CS2
1198 003154 013777 001162 175720 MOV UNNUM, @RSCS2 ; GET DRIVE NUMBER
1199 003162 032777 173577 17571C BIT #173577, @RSCS1 ; DID ALL BITS GET CLEARED
1200 003170 001401 BEQ +4 ; YES
1201 003172 104001 HLT ;CS1 ; NO, ALL BITS SHOULD BE 0
1202
1203 ;TEST BUS ADDRESS REGISTER
1204 003174 005777 175706 TST @RSBA ; IS BA REG. CLEARED
1205 003200 001401 BEQ +4 ; YES
1206 003202 104020 HLT !BA ; SHOULD BE 0
1207
1208 ;TEST DISK ADDRESS REGISTER
1209
1210 003204 005777 175700 TST @RSDA ; IS DA CLEARED
1211 003210 001401 BEQ +4 ; YES
1212 003212 104020 HLT !BA ; SHOULD BE 0
1213
1214 ;TEST ERROR REG RSER
1215
1216 003214 032777 177777 175672 BIT #177777, @RSER ; DID THESE BITS GET CLEARED
1217 003222 001401 BEQ +4 ; YES
1218 003224 104002 HLT !ER ; BITS(157015) SHOULD BE CLEARED
1219
1220 ;TEST RS MAINTENANCE REGISTER
1221 003226 032777 000077 175670 BIT #77, @RSMR ; DID THESE BITS GET CLEARED
1222 003234 001401 BEQ +4 ; YES
1223 003236 104220 HLT !MR ; BITS(77) SHOULD BE 0
1224
1225 ;TEST WC REG. IT SHOULD NOT CHANGE
1226 003240 022777 177777 175636 CMP #177777, @RSWC ; DID WC CHANGE
1227 003246 001401 BEQ +4 ; NO
1228 003250 104010 HLT !WC ; WHY DID IT CHANGE?

```

```

1229 ;*****
1230 ;TEST 3 SET AND CLEAR ALL REGISTERS
1231 ;*****
1232 003252 104400 †TST3: SCOPE
1233 ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1234 ;BITS 7,6,5,4,3,2&1
1235
1236 003254 104414 CLRDK ;CLEAR ALL RS REG
1237 003256 013737 001172 025404 MOV TIMSV,TIMES ;GET TIME
1238 003264 012777 003576 175606 MOV #3576,RSCS1 ;SET DISK FUNCTION BITS
1239 003272 022777 007776 175600 CMP #7776,RSCS1 ;ARE THESE BITS SET?
1240 003300 001401 BEQ +4 ;NO
1241 003302 104001 HLT !CS1 ;SHOULD = 3776
1242 003304 012777 002524 175566 MOV #2524,RSCS1 ;SET THESE BITS
1243 003312 022777 006724 175560 CMP #6724,RSCS1 ;DID THEY SET
1244 003320 001401 BEQ +4 ;YES
1245 003322 104001 HLT !CS1 ;SHOULD BE 2725
1246 003324 012777 001052 175546 MOV #1052,RSCS1 ;SET THESE BITS
1247 003332 022777 005252 175540 CMP #5252,RSCS1 ;ARE THEY =?
1248 003340 001401 BEQ +4 ;YES
1249 003342 104001 HLT !CS1 ;SHOULD = 1252
1250 003344 104400
1251 TST4: SCOPE
1252 ;CLEAR THE FUNCTION BITS
1253 003346 012777 043576 175524 MOV #43576,RSCS1 ;SET DISK FUNCTION BITS
1254 003354 005077 175520 CLR RSCS1
1255 003360 022777 004200 175512 CMP #4200,RSCS1 ;IS THE READY BIT SET
1256 003366 001401 BEQ +4 ;YES
1257 003370 104001 HLT !CS1 ;RSCS1 SHOULD = 4200
1258
1259 ;*****
1260 ;TEST 5 TEST RSCS2
1261 ;*****
1262 003372 104400 †TST5: SCOPE
1263
1264 003374 000005 RESET ;CLEAR WORLD
1265 003376 022777 000100 175476 CMP #100,RSCS2 ;DID THEY CLEAR?
1266 003404 001401 BEQ +4 ;YES
1267 003406 104200 HLT !CS2 ;NO
1268 003410 012777 021037 175464 MOV #21037,RSCS2 ;SET BITS 21017
1269 003416 022777 021137 175456 CMP #21137,RSCS2 ;DID THESE BITS GET SET
1270 003424 001405 BEQ 15 ;YES
1271 003426 017700 175450 MOV RSCS2,BAD
1272 003432 012701 021137 MOV #21137,GOOD ;WHAT CS2 SHOULD =
1273 003436 104000 HLT ;CS2 = BAD GOOD = CORRECT ANS

```

111

K03

MAINDEC-11-DZRSE-C
DZRSEC.P11

TSTS

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC
TEST RSCS2

MACY11 27(732) 25-SEP-76 10:44 PAGE 37

1274	003440	012777	020025	175434	15:	MOV	#20025, @RSCS2	; SET THESE BITS
1275	003446	022777	020125	175426		CMP	#20125, @RSCS2	; DID THESE BITS GET SET
1276	003454	001401				BEQ	.+4	; YES
1277	003456	104200				HLT	!CS2	; NO, CS2 SHOULD = 20125
1278	003460	012777	000012	175414		MOV	#12, @RSCS2	; LOAD THESE BITS
1279	003466	022777	000112	175406		CMP	#112, @RSCS2	; DID THESE BITS GET SET IN CS2
1280	003474	001401				BEQ	.+4	; YES
1281	003476	104200				HLT	!CS2	; BAD = CS2 GOOD = CORRECT ANS
1282	003500	012777	177777	175374		MOV	#-1, @RSCS2	; SET BITS
1283	003506	005077	175370			CLR	@RSCS2	; CLEAR THEM
1284	003512	022777	000100	175362		CMP	#100, @RSCS2	; DID CLEAR WORK
1285	003520	001401				BEQ	.+4	; YES
1286	003522	104200				HLT	!CS2	; R/W BITS DID NOT CLEAR
1287	003524	013777	001162	175350		MOV	UNNUM, @RSCS2	; GET UNIT #
1288	003532	104400						
1289					TST6:	SCOPE		
1290						; CAN WE	SET ALL THE RSBA BITS	
1291	003534	012777	177777	175344		MOV	#177777, @RSBA	; SET THE BITS
1292	003542	022777	177776	175336		CMP	#177776, @RSBA	; DID THEY SET
1293	003550	001401				BEQ	.+4	; YES
1294	003552	104020				HLT	!BA	; BITS 17776 SHOULD BE SET
1295	003554	012777	125252	175324		MOV	#125252, @RSBA	; SET THESE BITS
1296	003562	022777	125252	175316		CMP	#125252, @RSBA	; ARE THEY =
1297	003570	001401				BEQ	.+4	; YES
1298	003572	104020				HLT	!BA	; SHOULD BE 125252
1299	003574	012777	052524	175304		MOV	#52524, @RSBA	; SET THESE BITS
1300	003602	022777	052524	175276		CMP	#52524, @RSBA	; ARE THEY =
1301	003610	001401				BEQ	.+4	; YES
1302	003612	104020				HLT	!BA	; SHOULD BE 52524
1303								
1304	003614	104400						
1305					TST7:	SCOPE		
1306						; FLOAT A 1 THROUGH RSBA		
1307	003616	012701	000002		FLOTBA:	MOV	#2, GOOD	; GET A 2
1308	003622	000241				CLC		; CLEAR CARRY
1309	003624	010177	175256		15:	MOV	GOOD, @RSBA	; FLOAT NUMBER
1310	003630	017700	175252			MOV	@RSBA, BAD	; GET BA
1311	003634	020100				CMP	GOOD, BAD	; COMPARE BA
1312	003636	001401				BEQ	.+4	; BA CORRECT
1313	003640	104000				HLT		; BAD=BA GOOD=CORRECT ANS
1314	003642	006101				ROL	GOOD	; ROTATE NUMBER
1315	003644	103367				BCC	15	; LOOP TILL DONE

1316 003646 104400
1317
1318
1319
1320 003650 012777 177777 175230
1321 003656 005077 175224
1322 003662 005777 175220
1323 003666 001401
1324 003670 104020
1325 003672 104400
1326
1327
1328
1329 003674 012777 177777 175202
1330 003702 022777 177777 175174
1331 003710 001401
1332 003712 104010
1333 003714 012777 125252 175162
1334 003722 022777 125252 175154
1335 003730 001401
1336 003732 104010
1337 003734 012777 052525 175142
1338 003742 022777 052525 175134
1339 003750 001401
1340 003752 104010
1341 003754 104400
1342
1343
1344
1345 003756 012701 000001
1346 003762 000241
1347 003764 010177 175114
1348 003770 017700 175110
1349 003774 020100
1350 003776 001401
1351 004000 104000
1352 004002 006101
1353 004004 103367

TST10: SCOPE

;CLEAR THE RSBA REGISTER

```

MOV #177777, @RSBA ;SET RSBA EQUAL TO ALL ONES
CLR @RSBA
TST @RSBA ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
BEQ .+4 ;YES
HLT !BA ;NO
    
```

TST11: SCOPE

;CAN WE SET ALL BITS IN RSWC REGISTER

```

MOV #177777, @RSWC ;SET WC BITS
CMP #177777, @RSWC ;ARE ALL BITS SET
BEQ .+4 ;YES
HLT !WC ;NO
MOV #125252, @RSWC ;SET THESE BITS
CMP #125252, @RSWC ;ARE THEY =
BEQ .+4 ;YES
HLT !WC ;SHOULD BE 125252
MOV #52525, @RSWC ;SET THESE BITS
CMP #52525, @RSWC ;ARE THEY =
BEQ .+4 ;YES
HLT !WC ;SHOULD BE 152525
    
```

TST12: SCOPE

;FLOAT A 1 THROUGH RSWC

```

FLOTWC: MOV #1, GOOD ;GET A 1
        CLC ;CLEAR CARRY
1$:     MOV GOOD, @RSWC ;FLOAT NUMBER
        MOV @RSWC, BAD ;GET WC
        CMP GOOD, BAD ;COMPARE WC
        BEQ .+4 ;WC CORRECT
        HLT ;BAD=WC GOOD=CORRECT ANS
        ROL ;ROTATE NUMBER
        BCC 1$ ;LOOP TILL DONE
    
```

```

1354                                     ;CLEAR THE WORD COUNT REGISTER
1355 004006 104400 TST13: SCOPE
1356
1357 004010 012777 177777 175066      MOV    #177777, @RSWC ;SET RSWC REGISTER EQUAL TO ALL ONES
1358 004016 005077 175062              CLR    @RSWC
1359 004022 005777 175056              TST    @RSWC ;DID ALL BITS GET CLEARED
1360 004026 001401                      BEQ    .+4 ;YES
1361 004030 104010                      HLT    !WC ;NO
1362 004032 104400 TST14: SCOPE
1363
1364                                     ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1365
1366 004034 012777 177777 175046      MOV    #177777, @RSDA ;SET ALL BITS
1367 004042 022777 177777 175040      CMP    #177777, @RSDA ;ARE THE BITS SET
1368 004050 001401                      BEQ    .+4 ;YES
1369 004052 104004                      HLT    !DA ;NO
1370 004054 012777 125252 175026      MOV    #125252, @RSDA ;SET THESE BITS
1371 004062 022777 125252 175020      CMP    #125252, @RSDA ;ARE THEY =
1372 004070 001401                      BEQ    .+4 ;YES
1373 004072 104004                      HLT    !DA ;SHOULD BE 125252
1374 004074 012777 052525 175006      MOV    #52525, @RSDA ;SET THESE BITS
1375 004102 022777 052525 175000      CMP    #52525, @RSDA ;ARE THEY =
1376 004110 001401                      BEQ    .+4 ;YES
1377 004112 104004                      HLT    !DA ;SHOULD BE 52525
1378 004114 104400 TST15: SCOPE
1379
1380                                     ;FLOAT A 1 THROUGH RSDA
1381
1382 004116 012701 000001 FLOTDA: MOV    #1, GOOD ;GET A 1
1383 004122 000241          CLC    ;CLEAR CARRY
1384 004124 010177 174760 15: MOV    GOOD, @RSDA ;FLOAT NUMBER
1385 004130 017700 174754          MOV    @RSDA, BAD ;GET DA
1386 004134 020100          CMP    GOOD, BAD ;COMPARE DA
1387 004136 001401          BEQ    .+4 ;DA CORRECT
1388 004140 104000          HLT    ;BAD=DA GOOD=CORRECT ANS
1389 004142 006101          ROL    ;ROTATE NUMBER
1390 004144 103367          BCC    15 ;LOOP TILL DONE

```

N03

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST5

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 40
TEST RSCS2

```

1391          :CAN WE CLEAR THE RSDA REG.
1392 004146 104400 †TST16: SCOPE
1393
1394 004150 012777 177777 174732      MOV      #177777, @RSDA      ;SET RSDA TO ALL ONES
1395 004156 005077 174726              CLR      @RSDA              ;
1396 004162 005777 174722              TST      @RSDA              ;TEST FOR ZERO RSDA
1397 004166 001401                      BEQ      .+4                ;YES
1398 004170 104004                      HLT      !DA                ;ANS SHOULD BE 0
1399 004172 104400 TST17: SCOPE
1400
1401          ;SET AND CLEAR THE RSER REG.
1402
1403 004174 012777 177017 174712      MOV      #177017, @RSER      ;SET THESE BITS
1404 004202 022777 177017 174704      CMP      #177017, @RSER      ;DID THEY SET
1405 004210 001401                      BEQ      .+4                ;YES
1406 004212 104002                      HLT      !ER                ;RSER SHOULD = 157017
1407 004214 112777 000001 174672      MOVB     #1, @RSER          ;A MOVB INST
1408 004222 022777 000001 174664      CMP      #1, @RSER          ;SHOULD MODIFY COMPLETE WD
1409 004230 001401                      BEQ      .+4                ;OK
1410 004232 104002                      HLT      !ER
1411
1412 004234 104400 TST20: SCOPE
1413
1414 004236 012777 052005 174650      MOV      #52005, @RSER      ;SET THESE BITS
1415 004244 022777 052005 174642      CMP      #52005, @RSER      ;DID THEY SET
1416 004252 001401                      BEQ      .+4                ;YES
1417 004254 104002                      HLT      !ER                ;ER SHOULD = 52005
1418 004256 104400 TST21: SCOPE
1419
1420 004260 012777 125012 174626      MOV      #125012, @RSER     ;SET THESE BITS
1421 004266 022777 125012 174620      CMP      #125012, @RSER     ;DID THEY SET
1422 004274 001401                      BEQ      .+4                ;YES
1423 004276 104002                      HLT      !ER                ;ER SHOULD = 105012

```



```

1424 004300 104400          TST22: SCOPE
1425
1426 004302 012777 177017 174604      MOV      #177017, @RSER      ; SET THESE BITS
1427 004310 005077 174600              CLR      @RSER              ; CLEAR THEM
1428 004314 005777 174574              TST      @RSER              ; DID THEY CLEAR
1429 004320 001401              BEQ      .+4                 ; YES
1430 004322 104002              HLT      !ER                 ; SHOULD = 0
1431 004324 104400          TST23: SCOPE
1432
1433          ;SET AND CLEAR RSMR
1434
1435 004326 012777 000070 174570      MOV      #70, @RSMR          ; SET THESE BITS
1436 004334 017737 174564 001216      MOV      @RSMR, WORK        ; PUT INTO WORKABLE REG
1437 004342 042737 177700 001216      BIC      #177700, WORK      ; CLEAR JUNK
1438 004350 022737 000070 001216      CMP      #70, WORK          ; DID THEY SET
1439 004356 001401              BEQ      .+4                 ; YES
1440 004360 104220              HLT      !MR                 ; SHOULD = 70
1441 004362 104400          TST24: SCOPE
1442
1443 004364 012777 000070 174532      MOV      #70, @RSMR          ; SET BITS
1444 004372 005077 174526              CLR      @RSMR              ; CLEAR THEM
1445 004376 032777 000077 174520      BIT      #77, @RSMR         ; DID THEY CLEAR
1446 004404 001401              BEQ      .+4                 ; YES
1447 004406 104220              HLT      !MR                 ; BITS (77) SHOULD = 0
1448 004410 104400          TST25: SCOPE
1449
1450 004412 012777 000050 174504      MOV      #50, @RSMR         ; SET BITS
1451 004420 017737 174500 001216      MOV      @RSMR, WORK        ; PUT IN WORKABLE REG
1452 004426 042737 177700 001216      BIC      #177700, WORK      ; CLEAR JUNK
1453 004434 022737 000050 001216      CMP      #50, WORK          ; DID THESE BITS SET
1454 004442 001401              BEQ      .+4                 ; YES
1455 004444 104220              HLT      !MR                 ; BITS (50) SHOULD BE SET
1456 004446 104400          TST26: SCOPE
1457
1458 004450 012777 000020 174446      MOV      #20, @RSMR         ; SET BITS
1459 004456 017737 174442 001216      MOV      @RSMR, WORK        ; PUT INTO WORKABLE REG
1460 004464 042737 177700 001216      BIC      #177700, WORK      ; CLEAR JUNK
1461 004472 022737 000020 001216      CMP      #20, WORK          ; DID THEY SET
1462 004500 001401              BEQ      .+4                 ; YES
1463 004502 104220              HLT      !MR                 ; MR SHOULD AT LEAST HAVE A (21)

```

C04

MAINSEC-11-D2RSE-C
D2M.C.P11 15127

R511-R503 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 42
TEST 000 ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND DA

.....
TEST 27 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND DA
.....

464
465
466
467 004504 104400
468
469 004505 104414
470 004510 012777 007566 174362
471 004516 012777 000005 174410
472 004521 012777 006766 174346
473 004524 001401
474 004524 001401
475 004524 012777 000032 174334
476 004524 012777 006632 174326
477 004524 001401
478 004524 104400
479
480 004526 104400
481
482 004530 012777 001162 174314
483 004536 012777 177400 174306
484 004542 012777 001162
485 004548 000100
486 004554 012777 174266
487 004554 104400
488 004562 104400
489 004567 012777 005252 174252
490 004570 012777 000377 174210
491 004570 012777 177652 174206
492 004570 001401
493 004570 104010
494 004570 012777 000133 174224
495 004570 012777 177653 174216
496 004570 001401
497 004570 104010
498 004672 104400
499 004674 012777 001162 174204
500 004674 012777 001177 174232
501 004674 012777 177652 174170
502 004674 001401
503 004674 104010
504 004674 012777 000155 174186
505 004674 012777 177654 174188
506 004674 001401
507 004674 104020
508 004674 104414

TEST 27 SCOPE
BITST: CLDRK | CLEAR ALL RS REG
MOV 03516,DRSCS1 | LOAD CS1
MOV 05,DRSCS1 | LOAD BIT
CMP 06716,DRSCS1 | DID IT LOAD?
BEQ +4 | YES
HLT | CS1
MOV 032,DRSCS1
CMP 06632,DRSCS1
BEQ +4
HLT | CS1 | CS1 SHOULD = 6632

TEST 30: SCOPE
BITCS2: MOV UNUM,DRSCS2 | LOAD UNIT NUMBER
BIT 0177400,DRSCS2 | LOAD ALL BITS
CLRB DRSCS2 | CLR UPPER BYTE
MOV UNUM,COOD | GET UNIT NO.
BIT 0100,COOD | GET OR BIT
MOV DRSCS2,IND | GET CS2
CMP IND,COOD | IS CS2 CORRECT?
BEQ +4 | YES
HLT | LOAD BYTE DID NOT WORK

TEST 31: SCOPE
BITWC: MOV 0177652,DRWC | LOAD WC
MOV 0177,DRWC | LOAD BIT
CMP 0177652,DRWC | DID IT LOAD?
BEQ +4 | YES
HLT | WC | NO WC SHOULD = 177652
MOV 0123,DRWC
CMP 0177652,DRWC
BEQ +4
HLT | WC | WC SHOULD = 177653

TEST 32: SCOPE
BITDA: MOV 0327,DRDA | LOAD DA
MOV 0327,DRDA | LOAD BIT
CMP 0177652,DRDA | DID IT LOAD?
BEQ +4 | YES
HLT | DA | DA SHOULD = 177652
MOV 015,DRDA
CMP 0177654,DRDA
BEQ +4
HLT | DA | DA SHOULD = 177655
CLDRK | CLEAR ALL RS REG

```

1516 ;*****
1517 ;TEST 33 TEST DATA LATE IN CS2
1518 ;*****
1519 004744 104400 †TST33: SCOPE
1520
1521 ;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
1522
1523 SILOB: CLRDK ;CLEAR ALL RS REG
1524 004750 017700 174146 MOV @RSDB,BAD ;READ FROM EMPTY SILO
1525 004754 017700 174122 MOV @RS2,BAD ;GET CS2
1526 004760 012701 100100 MOV #100,00,GOOD ;GET CORRECT ANS
1527 004764 053701 001162 BIS UNNUM,000 ;FOR CS2
1528 004770 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1529 004772 001401 BEQ +4 ;YES
1530 004774 104200 HLT !CS2 ;SHOULD HAVE DLT ERROR
1531 004776 022777 144200 174074 CMP #144200,@RSCS1 ;DID SC AND TRE SET?
1532 005004 001401 BEQ +4 ;YES
1533 005006 104001 HLT !CS1 ;SC AND TRE SHOULD BE SET
1534 005010 012777 040000 174062 MOV #TRE,@RSCS1 ;CLEARS ERROR BIT
1535 005016 032777 140000 174054 BIT #140000,@RSCS1 ;DID SC + TRE CLEAR
1536 005024 001401 BEQ +4 ;YES
1537 005026 104001 HLT !CS1 ;TRE AND SC SHOULD BE 0
1538 005030 017700 174046 MOV @RSCS2,BAD ;GET CS2
1539 005034 042701 100000 BIC #BIT15,GOOD ;GET CORRECT ANS
1540 005040 020100 CMP GOOD,FAD ;IS CS2 CORRECT?
1541 005042 001401 BEQ +4 ;YES
1542 005044 104200 HLT !CS2 ;DLT SHOULD BE 0
1543 ;*****
1544 ;TEST 34 LOAD RSDB WITH ALL ONES AND ALL ZEROS
1545 ;*****
1546 005046 104400 †TST34: SCOPE
1547
1548 ZERONE: CLRDK ;CLEAR ALL RS REG
1549 005052 005077 174044 CLR @RSDB ;LOAD DB WITH ALL 0
1550 005056 012777 177777 174036 MOV #177777,@RSDB ;LOAD DB WITH ALL ONES
1551 005064 012737 002000 001216 MOV #2000,WORK ;TIME OUT ROUTINE
1552 005072 012701 000300 MOV #300,GOOD ;GET CORRECT FOR CS2
1553 005076 053701 001162 BIS UNNUM,GOOD
1554 005102 017700 173774 2$: MOV @RSCS2,BAD ;GET CS2
1555 005106 020100 CMP GOOD,BAD ;IS IT CORRECT?
1556 005110 001404 BEQ 3$ ;YES
1557 005112 005337 001216 DEC WORK ;TO WAIT FOR OR
1558 005116 001371 BNE 2$ ;TO SET
1559 005120 104200 HLT !CS2 ;OR SHOULD BE SET
1560 005122 005001 3$: CLR GOOD
1561 005124 017700 173772 MOV @RSDB,BAD ;LOAD BAD WITH DB
1562 005130 020100 CMP GOOD,BAD ;IS BAD CORRECT
1563 005132 001401 BEQ +4 ;YES
1564 005134 104000 HLT ;COULD NOT FLOAT 0 THROUGH DB
1565 005136 012701 177777 MOV #-1,GOOD ;LOAD GOOD WITH ANS
1566 005142 017700 173754 MOV @RSDB,BAD ;GET DATA FROM DB
1567 005146 020100 CMP GOOD,BAD ;IS DB CORRECT
1568 005150 001401 BEQ +4 ;YES
1569 005152 104000 HLT ;BAD SHOULD = 177777

```

```

1570 005154 104400          TST35: SCOPE
1571                                ;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION
1572
1573 005156 104414          SILO:  CLRDK                                ;CLEAR ALL RS REG
1574 005160 005001          CLR      R1                                ;CLEAR COUNTER
1575 005162 005201          15:  INC      R1                                ;INCREMENT COUNTER
1576 005164 010177 173732  MOV      R1,RSDB                            ;LOAD SILO
1577 005170 022701 000102  CMP      #66.,R1                            ;LAST LOC. YET?
1578 005174 001372          BNE      15                                ;NO LOOP AGAIN
1579 005176 012701 000200  MOV      #200,GOOD                          ;GET CORRECT ANS FOR CS2
1580 005202 053701 001162  BIS      UNNUM,GOOD
1581 005206 017700 173670  MOV      RS2CS2,BAD                          ;GET CS2
1582 005212 020100          CMP      GOOD,BAD                          ;IS CS2 CORRECT?
1583 005214 001401          BEQ      .+4                                ;YES
1584 005216 104200          HLT      !CS2                                ;OR SHOULD BE 1
1585 005220 005001          CLR      GOOD                                ;CLEAR LOCATION COUNTER
1586 005222 005201          25:  INC      GOOD                            ;ADD 1 TO IT
1587 005224 022701 000103  CMP      #67.,GOOD                          ;LAST LOC YET?
1588 005230 001405          BEQ      35                                ;YES
1589 005232 017700 173664  MOV      RSDB,BAD                          ;GET LOC FROM D3
1590 005236 020100          CMP      GOOD,BAD                          ;DO LOCATIONS MATCH?
1591 005240 001770          BEQ      25                                ;YES
1592 005242 104000          HLT      !CS2                                ;CAN NOT MATCH 66 LOCATIONS
1593 005244 032777 000200 173630 35:  BIT      #OR,RS2CS2                          ;IS OR 0
1594 005252 001401          BEQ      .+4                                ;YES
1595 005254 104200          HLT      !CS2                                ;OR SHOULD BE 0
1596
1597                                ;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR
1598
1599 005256 005001          45:  CLR      R1                                ;CLEAR COUNTER
1600 005260 005201          INC      R1                                ;ADD 1 TO COUNT
1601 005262 010177 173634  MOV      R1,RSDB                            ;PUT INTO COUNTER
1602 005266 022701 000103  CMP      #67.,R1                            ;DONE YET?
1603 005272 001401          BEQ      .+4                                ;YES
1604 005274 000771          BR      45                                ;NO DO AGAIN
1605 005276 032777 100000 173576  BIT      #DLT,RS2CS2                          ;DID DATA LATE SET?
1606 005304 001001          BNE      .+4                                ;YES
1607 005306 104200          HLT      !CS2                                ;DLT DID NOT SET
1608
1609                                ;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT
1610
1611 005310 017700 173606          MOV      RSDB,BAD
1612 005314 012701 000001          MOV      #1,GOOD
1613 005320 020100          CMP      GOOD,BAD
1614 005322 001401          BEQ      .+4
1615 005324 104000          HLT
1616 005326 104400          TST36: SCOPE

```



```

1671 :*****
1672 :TEST 40 MAINTENANCE TIMING TEST
1673 :*****
1674 005534 104400 TST40: SCOPE
1675
1676 :MODULE TESTED G092
1677 :THE FOLLOWING TEST ON THE RS03 DISK IS A SINGLE-STEPPED
1678 :MAINTENANCE MODE TEST ON THE RS03 TIMING LOGIC. THE ACTUAL
1679 :DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE RESISTER--I.E.
1680 :THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE
1681 :TIMING LOGIC. WE ARE TESTING THE ENTIRE TIMING TRACK LOGIC, INCLUDING INDEX,
1682 :PULSE FUNCTION, RESYNC AREA, SECTOR COUNTERS, ETC.
1683
1684 :PUT DRIVE IN MAINTENANCE MODE
1685 005536 104414 MRTIME: CLDK ;CLEAR DRIVE REGISTERS
1686 005540 052737 001040 001170 BIS #1040,ONCEE ;SET CLK CNT
1687 005546 104430 MRIND ;SEND INDEX PULSE TO MR REG
1688 005550 104420 MRCK ;CHECK MAINTENANCE REG FOR
1689 005552 022701 22701 ;22701
1690 005554 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1691 ;BY SENDING 2 CLOCK PULSES
1692 005556 104430 MRIND ;SEND MAINT INDEX PULSE
1693
1694 005560 104420 MRCK ;CHECK MAINT REG TO
1695 005562 022701 22701 ;EQUAL 22701
1696 005564 104000 HLT ;MR=BAD GOOD=CORRECTIONS
1697 ;COULD NOT INITIALIZE MR REG
1698 ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG
1699
1700 005566 005777 173326 TST @RSLA ;IS RSLA CLEARED
1701 005572 001401 BEQ +4 ;YES
1702 005574 104224 HLT !MR!LA ;RSLA SHOULD BE CLEARED
1703 ;WITH THE INDEX PULSE
1704
1705 :PERFORM MAINTENANCE CLOCK OPERATION 512 TIMES TO
1706 :PROVIDE CLOCK TO STEP TIMING THRU RESYNC PERIOD
1707 :IF SECTOR PULSE IS ASSERTED DURING THIS LOOP
1708 :CHECK SECTOR BOUNDARY COUNTER AND E12
1709
1710 005576 012737 001000 001204 MOV #512.,REPT
1711 005604 104446 MRTIM1: MCLK1 ;CLOCK MAINT REG WITH AN 11
1712 005606 104420 MRCK ;CHECK MR REG TO
1713 005610 032711 32711 ;EQUAL 32711
1714 005612 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1715 005614 104450 MCLK0 ;CLOCK MR WITH A 1
1716 005616 104420 MRCK ;CHECK MR TO
1717 005620 022701 22701 ;EQUAL 22701
1718 005622 104000 HLT ;BAD=MR REG GOOD=CORRECTIONS
1719 005624 005337 001204 DEC REPT ;IS THE LOOP DONE YET?
1720 005630 001365 BNE MRTIM1 ;NO-LOOP

```

```

1721 ;AFTER ONE MORE CLOCK, SECTOR PULSE SHOULD BE ASSERTED
1722 ;IF NOT, CHECK SECTOR BOUNDARY COUNTER, SECTOR BOUNDARY FF (E21) AND Z12
1723
1724 005632 104446 MCLK1 ;CLOCK MAINT REG WITH AN 11
1725 005634 104420 MRCK ;CHECK MR REG TO
1726 005636 032311 32311 ;EQUAL 32311
1727 005640 104000 HLT ;MR=BAD GOOD=CORRECTIONS
1728 005642 104450 MCLK0 ;CLOCK MR WITH A 1
1729 005644 104420 MRCK ;CHECK MAINT REG
1730 005646 022301 22301 ;TO EQUAL 22301
1731 005650 104000 HLT ;MR=BAD GOOD-CORRECT ANS
1732 005652 005777 173242 TST @RSLA ;DOES LOOK AHEAD REG=0
1733 005656 001401 BEQ MRT2 ;YES-CONT
1734 005660 104224 HLT !MR!LA ;LOOK AHEAD REG SHOULD=0
1735 ;PERFORM MAINTENANCE CLOCK OPERATION 40 TIMES TO PROVIDE
1736 ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA
1737
1738 005662 005002 MRT2: CLR R2 ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
1739 005664 012737 000050 001204 MOV #40.,REPT ;40 CLOCKS TO STEP THRU PRE-AMBLE
1740 005672 104446 MRT2A: MCLK1 ;CLOCK MR WITH AN 11
1741 005674 104420 MRCK ;CHECK MAINT REG
1742 005676 033711 33711 ;EQUAL 33711
1743 005700 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1744 005702 104450 MCLK0 ;CLOCK MR REG WITH A 1
1745 005704 104420 MRCK ;CHECK MR REG
1746 005706 023701 23701 ;TO EQUAL 23701
1747 005710 104000 HLT ;MR = BAD GOOD = CORRECTANS
1748 005712 005337 001204 DEC REPT ;REPEAT
1749 005716 001365 BNE MRT2A ;LOOP 40 TIMES
1750
1751 ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
1752 005720 012737 002200 001204 MOV #18.*64.,REPT ;18 CLOCKS PER DATA WORD
1753 005726 104446 MRT2B: MCLK1 ;CLOCK MR WITH AN 11
1754 005730 104420 MRCK ;CHECK MAINT REG
1755 005732 033711 33711 ;TO EQUAL 33711
1756 005734 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1757 005736 104450 MCLK0 ;CLOCK MR REG WITH A 1
1758 005740 104420 MRCK ;CHECK MR REG
1759 005742 023701 23701 ;TO EQUAL 23701
1760 005744 104000 HLT ;MR=BAD GOOD=CORRECTANS
1761 005746 005337 001204 DEC REPT ;REPEAT
1762 005752 001365 BNE MRT2B ;LOOP

```

```

1763          :SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
1764          :AND THE DEAD BAND ON THE SECTOR
1765
1766 005754 012737 000214 001204  MRT2C:  MOV      #140.,REPT      ;AMOUNT OF CLOCKS TO END OF SECTOR
1767 005762 104446          MCLK1      ;CLOCK MR WITH AN 11
1768 005764 104420          MRCK        ;CHECK MAINT REG
1769 005766 033711          33711      ;TO EQUAL 33711
1770 005770 104000          HLT        ;MR = BAD GOOD = CORRECT ANS
1771 005772 104450          MCLKD     ;CLOCK MR REG WITH A 1
1772 005774 104420          MRCK        ;CHECK MAINT REG
1773 005776 023701          23701      ;TO EQUAL 23701
1774 006000 104000          HLT        ;MR=BAD GOOD=CORRECT ANS
1775 006002 005337 001204  DEC      REPT      ;REPEAT
1776 006006 001365          BNE      MRT2C     ;LOOP
1777 006010 104446          MCLK1      ;CLOCK MR REG WITH 11
1778 006012 104420          MRCK        ;CHECK MR REG
1779 006014 033711          33711      ;TO EQUAL 33711
1780 006016 104000          HLT        ;MR = BAD GOOD = CORRECT ANS
1781          :ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
1782          :IF NOT, CHECK E16-6
1783
1784 006020 104450          MCLKD     ;CLOCK MR WITH A 1
1785 006022 104420          MRCK        ;MAINT REG SHOULD
1786 006024 023701          23701      ;EQUAL 23701
1787 006026 104000          HLT        ;MR=BAD GOOD=CORRECT ANS
1788 006030 104446          MCLK1      ;CLOCK MR WITH AN 11
1789 006032 104420          MRCK        ;CHECK MAINT REG
1790 006034 032311          32311      ;SHOULD EQUAL 32311
1791 006036 104000          HLT        ;MR=BAD GOOD=CORRECT ANS
1792
1793          :LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)
1794
1795 006040 022777 000000 173060  CMP      #0,ARSDT   ;INTERLEAVED?
1796 006046 001403          BEQ      3$        ;NO
1797 006050 062702 004000          ADD      #4000,R2  ;YES
1798 006054 000402          BR       2$        ;CONT
1799 006056 062702 000100 3$:  ADD      #100,R2   ;INCREMENT SECTOR COMPARE
1800 006062 020277 173032 2$:  CMP      R2,ARSLA  ;LA REG SHOULD=100
1801 006066 001401          BEQ      1$        ;LA IS CORRECT
1802 006070 104224          HLT      !MR!LA   ;LA SHOULD=100

```



```

1803 ;REPEAT NEXT STEPS 62 TIMES. LOOK-AHEAD REGISTER SHOULD INCREMENT
1804 ;TO SHOW NEXT SECTOR. CHECK FOR ALL SECTORS. IF DRIVE IS NOT
1805 ;INTERLEAVED, LA = 200,300, ETC. IF DRIVE IS INTERLEAVED,
1806 ;LA = 100, 4100, 200, 4200 ETC. SEE SERVICE MANUAL FOR DETAILS.
1807
1808 006072 012737 000076 001206 1S: MOV #62.,REPT1
1809 006100 012737 002465 001204 MRT3: MOV #1333.,REPT
1810 006106 104452 3S: MCLKB ;CLOCK MR WITH A 1 AND A 11
1811 006110 005337 001204 DEC REPT ;STEP THROUGH
1812 006114 001374 BNE 3S ;SECTOR
1813 006116 104450 MCLKO ;CLOCK MR WITH A 1
1814 006120 104420 MRCK ;MAINT REG
1815 006122 022701 22701 ;SHOULD EQUAL 22701
1816 006124 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1817 006126 104446 MCLK1 ;1 MORE CLK ASSERTS SECTOR PULSE
1818 006130 104420 MRCK ;MAINT REG SHOULD
1819 006132 032311 32311 ;EQUAL 32311
1820 006134 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1821 006136 022777 000000 172762 CMP #0,ARSDT ;DRIVE INTERLEAVED?
1822 006144 001420 BEQ 6S ;YES
1823 006146 032737 001000 001170 BIT #BIT9,ONCEE ;DO I SET 4000
1824 006154 001406 BEQ 4S ;OR CLEAR IT IN RSLA
1825 006156 042737 001000 001170 BIC #BIT9,ONCEE
1826 006164 162702 004000 SUB #4000,R2
1827 006170 000406 BR 6S
1828 006172 052737 001000 001170 4S: BIS #BIT9,ONCEE
1829 006200 062702 004000 ADD #4000,R2
1830 006204 000402 BR 5S
1831 006206 062702 000100 6S: ADD #100,R2 ;INCREMENT SECTOR COMPARE
1832 006212 017700 172702 5S: MOV ARSLA,BAD ;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
1833 006216 010201 MOV R2,GOOD ;GET CORRECT ANS FOR RSLA
1834 006220 020100 CMP GOOD,BAD ;COMPARE FOR CORRECT ANS
1835 006222 001401 BEQ 1S ;RSLA IS GOOD
1836 006224 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
1837
1838 006226 005337 001206 1S: DEC REPT1 ;REPEAT 62
1839 006232 001322 MRT3 ;TIMES
1840 006234 012737 002465 001204 MOV #1333.,REPT ;COUNT FOR LAST SECTOR
1841 006242 104452 2S: MCLKB ;CLOCK
1842 006244 005337 001204 DEC REPT ;THRU
1843 006250 001374 BNE 2S ;LAST SECTOR
1844 006252 017700 172642 MOV ARSLA,BAD ;GET CONTENTS OF RSLA
1845 006256 012701 007777 MOV #7777,GOOD ;GET CORRECT ANS
1846 006262 020100 CMP GOOD,BAD ;DOES RSLA EQUAL 7777
1847 006264 001401 BEQ .+4 ;YES
1848 006266 104000 HLT ;BAD=RSLA GOOD=CORRECT ANS

```

```

1849 ;**** *****
1850 ;TEST 1 SECTOR FRACTION TEST
1851 ;**** *****
1852 006270 104400 TST41: SCOPE

```

```

1853 ;MODULE TESTED G092
1854 ;CLOCK THROUGH AN ENTIRE TRACK IN MAINT MODE WHILE
1855 ;CHECKING FOR THE PROPER OPERATION OF THE SECTOR FRACTION IN THE LOOK-AHEAD REG.
1856 ;WHEN THE LAST WORD IS BEING TRANSFERRED, SECTOR AND FRACTION
1857 ;IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK --
1858 ;HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL
1859 ;CLEAR THE FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA
1860 ;SHOULD INDICATE 7700 ON ANOTHER MAINTENANCE CLOCK.

```

```

1863 006272 104414 MRT4: CLDK ;CLEAR DRIVE REGISTERS
1864 006274 052737 000040 001170 BIS #40,ONCEE ;SET FLAG BITS
1865 006302 042737 003000 001170 BIC #3000,ONCEE
1866 006310 005037 001176 CLR MCCNT ;CLEAR MAINT CLOCK COUNTER
1867 006314 005002 CLR R2 ;CLEAR R2 FOR SECTOR COUNTER
1868 006316 104430 MRIND ;SEND INDEX PULSE TO MR REG
1869 006320 104420 MRCK ;CHECK MAINTENANCE REG FOR
1870 006322 022701 22701 ;22701
1871 006324 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1872 ;BY SENDING 2 CLOCK PULSES
1873 006326 104430 MRIND ;ISSUE A MAINT INDEX PULSE
1874 ;TO CLEAR THE DRIVE
1875 006330 104420 MRCK ;CHECK MAINT REG
1876 006332 022701 22701 ;TO EQUAL 22701
1877 006334 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```

```

1878 ;ISSUE 512 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA
1879
1880
1881 006336 012737 001000 001204 MRT4A: MOV #512.,REPT ;COUNT TO STEP THRU RESYNC AREA
1882 006344 104446 MCLK1 ;CLOCK THROUGH RESYNC
1883 006346 104420 MRCK ;CHECK MAINT REG
1884 006350 032711 32711 ;TO EQUAL 32711
1885 006352 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1886 006354 104450 MCLK0 ;CLOCK MR REG
1887 006356 104420 MRCK ;CHECK MR REG
1888 006360 022701 22701 ;TO EQUAL 22701
1889 006362 104000 HLT ;BAD=MR GOOD=CORRECT ANS
1890 006364 022777 000000 172526 CMP #0,RSLA ;LOOK AHEAD REG
1891 006372 001401 BEQ +4 ;EQUAL 0
1892 006374 104204 HLT !LA
1893 006376 005337 001204 DEC REPT ;LOOP THROUGH
1894 006402 001360 BNE MRT4A ;RESYNC AREA

```

```

1895 ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE
1896
1897
1898 006404 104446 MCLK1 ;CLOCK MR WITH AN 11
1899 006406 104420 MRCK ;CHECK MAINT REG FOR SECTOR PULSE
1900 006410 032311 32311 ;MR SHOULD=32311
1901 006412 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```

```

1902 006414 104450      MRT4B:  MCLKD      ;CLOCK MR REG WITH A 1
1903 006416 104420      MRCK      ;CHECK MAINT REG
1904 006420 022301      22301    ;TO EQUAL 22301
1905 006422 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
1906
1907      ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)
1908
1909 006424 017700 172470      MOV      @RSLA,BAD ;GET RSLA
1910 006430 010201      MOV      R2,GOOD  ;GET CORRECT ANS
1911 006432 020100      CMP      GOOD,BAD ;IS THE RSLA REG CORRECT
1912 006434 001401      BEQ      1$      ;YES
1913 006436 104000      HLT      ;RSLA=BAD GOOD=CORRECTANS
1914
1915      ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
1916      ;AREA WHILE CHECKING THE SECTOR FRACTION
1917
1918 006440 012737 000122 001204 1$:  MOV      #82.,REPT ;FOR FIRST FRACTION CHANGE
1919 006446 104422      MRT4C:  MRCLK      ;CLOCK MR REG WITH AN 11 AND A 1
1920 006450 017700 172444      MOV      @RSLA,BAD ;GET RSLA
1921 006454 010201      MOV      R2,GOOD  ;GET CORRECT ANS
1922 006456 020001      CMP      BAD,GOOD ;IS RSLA CORRECT
1923 006460 001401      BEQ      1$      ;YES
1924 006462 104000      HLT      ;BAD=RSLA GOOD=CORRECT ANS
1925 006464 005337 001204 1$:  DEC      REPT     ;LOOP ON
1926 006470 001366      BNE      MRT4C   ;PREAMBLE AREA
1927
1928      ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1929
1930 006472 104422      MRCLK      ;CLOCK MR WITH AN 11 AND A 1
1931 006474 005202      INC      R2      ;COUNT THE FRACTION
1932 006476 017700 172416      MOV      @RSLA,BAD ;GET RSLA
1933 006502 010201      MOV      R2,GOOD  ;GET CORRECT ANS
1934 006504 020001      CMP      BAD,GOOD ;IS RSLA CORRECT?
1935 006506 001401      BEQ      2$      ;YES
1936 006510 104000      HLT      ;RSLA=BAD GOOD=CORRECT ANS
1937
1938      ;FIRST FRACTION CHANGES AFTER 82 MAINT CLKS, THE REST
1939      ;CHANGE AFTER 20 MAINTENANCE CLOCKS
1940
1941 006512 012737 000076 001204 2$:  MOV      #62.,REPT ;COUNT FOR WORDS IN A SECTOR
1942 006520 012737 000023 001206 MRT4D:  MOV      #19.,REPT1 ;COUNT FOR SECT FRACT TO CHANGE
1943 006526 104422      MRT4E:  MRCLK      ;CLOCK MR WITH AN 11 AND A 1
1944 006530 017700 172364      MOV      @RSLA,BAD ;GET RSLA
1945 006534 010201      MOV      R2,GOOD  ;GET CORRECT ANS
1946 006536 020100      CMP      GOOD,BAD ;IS RSLA CORRECT?
1947 006540 001401      BEQ      1$      ;YES
1948 006542 104000      HLT      ;RSLA=BAD GOOD=CORRECT ANS
1949 006544 005337 001206 1$:  DEC      REPT1   ;LOOP
1950 006550 001366      BNE      MRT4E

```

```

;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1951
1952
1953 006552 104422 MRCLK ;CLOCK MR WITH AN 11 AND A 1
1954 006554 022702 007777 CMP #7777,R2 ;AT THE LAST SECTOR-LAST FRACTION?
1955 006560 001472 BEQ MRT4F ;YES, FINISH THE SECTOR
1956 006562 005202 INC R2 ;NO, ADD 1 TO FRACTION
1957 006564 017700 172330 4S: MOV @RSLA,BAD ;GET RSLA
1958 006570 022777 000000 172330 CMP #0,@RSDT ;IS THIS DRIVE INTERLEAVED?
1959 006576 001431 BEQ 12$ ;NO
1960 006600 032737 002000 001170 BIT #BIT10,ONCEE ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
1961 006606 001425 BEQ 12$ ;NO
1962 ;RSLA NOW POINTS TO NEXT INTERLEAVED
1963 ;SECTOR; BIT 9 IN ONCEE INDICATES
1964 ;WHETHER RSLA SHOULD NOW
1965 ;BE BETWEEN 0000-3700(1)
1966 ;OR 4000-7700(0).
1967 006610 032737 001000 001170 BIT #BIT9,ONCEE ;SHOULD RSLA BE BETWEEN 0000-3700?
1968 006616 001004 BNE 9$ ;YES
1969 006620 052737 001000 001170 BIS #BIT9,ONCEE ;SET FOR NEXT PASS
1970 006626 000406 BR 10$
1971 006630 042737 001000 001170 9S: BIC #BIT9,ONCEE ;CLEAR FOR NEXT PASS
1972 006636 042702 004000 BIC #4000,R2 ;MAKE RSLA LESS THAN 4000
1973 006642 000404 BR 5$
1974 006644 062702 004000 10$: ADD #4000,R2 ;COMPENSATE FOR
1975 006650 162702 000100 SUB #100,R2 ;INTERLEAVING
1976 006654 042737 002000 001170 5$: BIC #BIT10,ONCEE ;CLEAR FLAG FOR NEXT SECTOR
1977 006662 010201 12$: MOV R2,GOOD ;GET CORRECT ANSWER FOR RSLA
1978 006664 020100 CMP GOOD,BAD ;IS RSLA CORRECT
1979 006666 001401 BEQ 2$ ;YES
1980 006670 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
1981 006672 005337 001204 2$: DEC REPT ;HAS SECTOR FRACTION REACHED 77?
1982 006676 001310 BNE MRT4D ;NO
1983
1984 ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT
1985
1986 006700 010203 11$: MOV R2,R3
1987 006702 042703 177700 BIC #177700,R3 ;CHECK SECTOR FRACTION
1988 006706 022703 000077 CMP #77,R3 ;END OF SECTOR?
1989 006712 001402 BEQ 3$ ;YES
1990 006714 000137 006414 JMP MRT4B ;NO, BEGINNING OF NEXT
1991 006720 012737 000012 001206 3$: MOV #10,REPT1 ;SETUP LOOP TO FINISH
1992 006726 012737 000001 001204 MOV #1,REPT ;THIS SECTOR
1993 006734 052737 002000 001170 BIS #BIT10,ONCEE ;REPT HAS GONE TO ZERO FOR THIS SECTOR
1994 006742 000137 006526 JMP MRT4E ;LOOP
1995
1996 006746 012737 000010 001204 MRT4F: MOV #8,REPT
1997 006754 104422 1$: MRCLK ;CLOCK MR WITH AN 11 AND A 1
1998 006756 017700 172136 MOV @RSLA,BAD ;GET RSLA
1999 006762 010201 MOV R2,GOOD ;R2 SHOULD=7777
2000 006764 020100 CMP GOOD,BAD ;IS RSLA CORRECT-END OF DISK?
2001 006766 001401 BEQ 2$ ;YES
2002 006770 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS (7777)
2003 006772 005337 001204 2$: DEC REPT ;FINISH
2004 006776 001366 BNE 1$ ;LOOP

```

N04

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST41

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 53
SECTOR FRACTION TEST

```
2005 ;SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
2006 ;RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.
2007
2008 007000 104422 MRT4G: MRCLK ;CLOCK MR WITH AN 11 AND A 1
2009 007002 017700 172112 MOV @RSLA,BAD ;GET RSLA
2010 007006 012701 007700 MOV #7700,GOOD ;GET CORRECT ANS
2011 007012 020100 CMP GOOD,BAD ;IS RSLA CORRECT?
2012 007014 001401 BEQ 1$ ;YES
2013 007016 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
2014 007020 104430 1$: MRIND ;ISSUE AN INDEX PULSE TO
2015 ;CLEAR THE DRIVE
2016 007022 017700 172072 MOV @RSLA,BAD ;GET RSLA
2017 007026 005001 CLR GOOD ;GET CORRECT ANS
2018 007030 020100 CMP GOOD,BAD ;IS RSLA CORRECT?
2019 007032 001401 BEQ 2$ ;YES
2020 007034 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
2021 007036 104420 2$: MRCK ;CHECK MR REG
2022 007040 022701 22701 ;TO EQUAL 22701
2023 007042 104000 HLT ;MR=BAD GOOD=CORRECT ANS
```

B05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST42

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC
ILLEGAL FUNCTION TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 54

```

2024 :*****
2025 :TEST 42 ILLEGAL FUNCTION TEST
2026 :*****
2027 007044 104400 †TST42: SCOPE
2028
2029 :MODULE TESTED M7759, M7770
2030 :TEST ILLEGAL FUNCTION (ILF) IN RSER. SEND AN ILLEGAL FUNCTION
2031 :CODE TO THE DRIVE CONTROL REGISTER WITHOUT SETTING THE GO BIT.
2032 :THE "ILF" BIT SHOULD NOT BE SET. THE "GO" BIT IS THEN SET. A
2033 :CHECK IS THEN MADE FOR "ATA" AND "ERR" TO BE SET
2034 :IN THE DRIVE STATUS REGISTER (RSDS) AND "ILF" IN THE DRIVE ERROR
2035 :REGISTER (RSER). ALL ILLEGAL FUNCTION CODES ARE CHECKED.
2036 :ILLEGAL FUNCTIONS ARE DETECTED ON M7759 BY E20-B
2037
2038 007046 104414 MRILF: CLRDK ;CLEAR ALL THE DRIVE REGISTERS
2039 007050 042737 000040 001170 BIC #BITS,ONCEE ;CLEAR CLOCK CNT FLAG
2040 (1)7056 032737 000002 001170 BIT #BIT1,ONCEE ;WAS THERE AN ERROR
2041 007064 001002 BNE MRLF1 ;YES DO NOT CHANGE "ILF" CODE
2042 007066 012702 000003 MOV #3,R2 ;SETUP FIRST "ILF" CODE
2043 ;PUT DRIVE IN MAINTENANCE MODE
2044
2045 007072 104416 MRLF1: MRDMD ;PUT DRIVE INTO MAINT MODE
2046 007074 104420 MRCK ;CHECK MR REG TO
2047 007076 022701 22701 ;EQUAL 22701
2048 007100 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2049
2050 ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE
2051
2052 007102 104430 MRIND
2053 007104 010277 171770 MRLF2: MOV R2,RSCS1 ;SEND "ILF" WITH THE "GO" BIT
2054 007110 017700 171776 MOV RSDS,BAD ;GET DRIVE STATUS REG
2055 007114 012701 150600 MOV #150600,GOOD ;GET CORRECT ANS
2056 007120 020100 CMP GOOD,BAD ;IS RSDS CORRECT?
2057 007122 001440 BEQ IS ;YES
2058 007124 104402 007130 TYPE #+2 ;ASCIZ (15)(12)"ILLEGAL FUNCTION CODE SENT TO DRIVE="
2059 007200 010237 001216 MOV R2,WORK ;GET FUNCTION CODE
2060 007204 013746 001216 MOV WORK,-(6) ;PUT WORK ON STACK
2061 007210 104406 TYPES ;TYPE STACK IN OCTAL - SUPPRESS
2062 007212 052737 000002 001170 BIS #BIT1,ONCEE ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
2063 007220 104000 HLT ;RSDS=BAD GOOD=CORRECT ANS
2064 007222 104040 HLT :DS
2065
2066 007224 042737 000002 001170 1$: BIC #BIT1,ONCEE ;CLEAR ERROR FLAG
2067 007232 017700 171656 MOV RRSER,BAD ;GET RSER
2068 007236 012701 000001 MOV #1,GOOD ;GET CORRECT ANS
2069 007242 020100 CMP GOOD,BAD ;DID "ILF" SET IN RSER
2070 007244 001404 BEQ 2$ ;YES
2071 007246 052737 000002 001170 BIS #BIT1,ONCEE ;SET ERROR BIT
2072 007254 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
2073 007256 042737 000002 001170 2$: BIC #BIT1,ONCEE ;CLEAR ERROR FLAG

```



```

2111 :*****
2112 :TEST 43 TEST NO-OP CODES 1 AND 21
2113 :*****
2114 007566 104400 TST43: SCOPE
2115
2116 ;MODULE TESTED M7759
2117 007570 104414 MROP: CLRDK ;CLEAR ALL DRIVE REGISTERS
2118 007572 042737 000004 001170 BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
2119 007600 104416 MROMD ;PUT DRIVE INTO MAINT MODE
2120 007602 104420 MRCK ;CHECK MR REG TO
2121 007604 022701 22701 ;EQUAL 22701
2122 007606 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2123 007610 032737 000010 001170 BIT #BIT3,ONCEE ;TESTING CODE 1
2124 007616 001031 BNE 3$ ;NO CODE 21
2125 007620 012777 000001 171252 MOV #1,RSRCS1 ;LOAD NO-OP FUNCTION
2126 007626 012737 040001 001216 MOV #1,WORK ;LOAD NO-OP FUNCTION
2127 007634 005777 171254 TST RSRER ;ANY ERRORS
2128 007640 001403 BEQ 1$ ;NO
2129 007642 004737 022026 JSR PC,NOPERR ;TYPE IT
2130 007646 104040 HLT :DS ;TYPE ERROR
2131 007650 022777 010600 171234 1$: CMP #10600,RSRDS ;IS RSDS CORRECT
2132 007656 001403 BEQ 2$ ;YES
2133 007660 004737 022026 JSR PC,NOPERR ;RSDS SHOULD
2134 007664 104040 HLT :DS ;EQUAL 10600
2135 007666 042737 000004 001.70 2$: BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
2136
2137 ;TEST NO-OP FUNCTION CODE 21
2138
2139 007674 052737 000010 001170 BIS #BIT3,ONCEE ;TEST TESTING CODE 21 FLAG
2140 007702 012737 000021 001216 3$: MOV #21,WORK ;LOAD CODE 21
2141 007710 012777 000021 171162 MOV #21,RSRCS1 ;LOAD FUNCTION
2142 007716 005777 171172 TST RSRER ;ANY ERRORS?
2143 007722 001403 BEQ 4$ ;NO
2144 007724 004737 022026 JSR PC,NOPERR ;YES, TYPE ERROR
2145 007730 104040 HLT :DS ;ERROR DURING NO-OP FUNCTION
2146 007732 022777 010600 171152 4$: CMP #10600,RSRDS ;IS RSDS CORRECT
2147 007740 001403 BEQ 5$ ;YES
2148 007742 004737 022026 JSR PC,NOPERR ;TYPE ERROR
2149 007746 104040 HLT :DS ;RSDS SHOULD=10600
2150 007750 042737 000014 001170 5$: BIC #14,ONCEE ;CLEAR TEST BITS

```


E05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST44

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 57
TEST NO-OP FUNCTION WITH ERROR BITS SET

```

2151 ;*****
2152 ;TEST 44 TEST NO-OP FUNCTION WITH ERROR BITS SET
2153 ;*****
2154 007756 104400 †TST44: SCOPE
2155
2156 ;MODULE TESTED M7759
2157 007760 104414 MROPER: CLRDK ;CLEAR ALL REGISTERS
2158 007762 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2159 007764 104420 MRCK ;CHECK MR REG
2160 007766 022701 22701 ;TO EQUAL 22701
2161 007770 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2162 007772 104430 MRIND ;SEND INDEX PULSE
2163
2164 007774 012777 177777 171112 MOV #-1,RSER ;LOAD RSER WITH ERRORS
2165 010002 013701 001166 MOV UNCMP,GOOD ;GET DRIVE UNDER TEST
2166 010006 042701 177400 BIC #177400,GOOD
2167 010012 017700 171100 MOV RSAS,BAD ;GET RSAS REG
2168 010016 020100 CMP GOOD,BAD ;DID ATA BIT SET CAUSED BY ERROR
2169 010020 001427 BEQ 1$ ;YES
2170 010022 104402 010026 TYPE ;ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
2171 010076 104000 HLT ;RSAS=BAD GOOD=CORRECT ANS
2172 010100 012737 000001 001216 1$: MOV #1,WORK ;SETUP FOR NO-OP CODE 1
2173 010106 032737 000010 001170 BIT #BIT3,ONCEE ;TESTING CODE 1
2174 010114 001004 BNE 2$ ;YES
2175 010116 012777 000001 170754 MOV #1,RSRCSI ;SEND NO-OP CODE 1
2176 010124 000406 BR 3$ ;CHECK FOR ERRORS
2177 010126 012737 000021 001216 2$: MOV #21,WORK ;SETUP FOR CODE 21
2178 010134 012777 000021 170736 MOV #21,RSRCSI ;SENT NO-OP CODE 21
2179 010142 017700 170746 3$: MOV RSER,BAD ;GET RSER REG
2180 010146 012701 177017 MOV #177017,GOOD ;GET CORRECT ANS
2181 010152 020100 CMP GOOD,BAD ;DID RSER CHANGE WITH NO-OP
2182 010154 001411 BEQ 4$ ;NO
2183 010156 104402 010162 TYPE ;ASCIZ <15><12>"RSER "
2184 010172 004737 022122 JSR PC,CHG
2185 010176 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
2186 010200 017700 170712 4$: MOV RSAS,BAD ;GET RSAS
2187 010204 013701 001166 MOV UNCMP,GOOD ;GET CORRECT ANS
2188 010210 042701 177400 BIC #177400,GOOD ;CLEAR J.L.K
2189 010214 020100 CMP GOOD,BAD ;IS RSAS CORRECT
2190 010216 001411 BEQ 5$ ;YES
2191 010220 104402 010224 TYPE ;ASCIZ <15><12>"RSAS "
2192 010234 004737 022122 JSR PC,CHG ;TYPE ERROR
2193 010240 104000 HLT ;RSAS=BAD GOOD=CORRECT ANS
2194 010242 017700 170644 5$: MOV RSAS,BAD ;GET RSAS
2195 010246 012701 150600 MOV #150600,GOOD ;GET CORRECT ANS
2196 010252 020100 CMP GOOD,BAD ;DID RSAS CHANGE
2197 010254 001411 BEQ 6$ ;NO
2198 010256 104402 010262 TYPE ;ASCIZ <15><12>"RSAS "
2199 010272 004737 022122 JSR PC,CHG ;TYPE ERROR
2200 010276 104000 HLT ;RSAS=BAD GOOD=CORRECT ANS
2201 010300 032737 000010 001170 6$: BIT #BIT3,ONCEE ;TESTING CODE 21
2202 010306 001005 BNE 7$ ;YES, GET OUT
2203 010310 052737 000010 001170 BIS #BIT3,ONCEE ;SET CODE 21 FLAG
2204 010316 000137 007760 JMP MROPER ;TEST CODE 21
2205 010322 042737 000010 001170 7$: BIC #BIT3,ONCEE ;DONE CLEAR FLAG AND CONT.

```

F05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST45

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 58
BLOCK SEARCH TEST 1

```

2206 :*****
2207 :TEST 45          BLOCK SEARCH TEST 1
2208 :*****
2209 010330 104400 TST45: SCOPE
2210
2211 :MODULE TESTED: M7759 M7754 M7771 M7770
2212 :A DRIVE SEARCH FUNCTION IS GIVEN TO THE DRIVE FOR SECTOR 3.
2213 : (SECTOR 41 IF SECTOR INTERLEAVING IS ENABLED) THE
2214 : POSITIONING IN PROGRESS BIT (PIP) AND THE DRIVE READY BIT
2215 : (DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE
2216 : ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.
2217
2218 010332 104414 MRSRCH: CLRDK          ;CLEAR ALL REGISTERS
2219 010334 052737 000040 001170 BIS          #BITS,ONCEE ;SET CLOCK FLAG
2220 010342 104416 MROMD          ;PUT DRIVE INTO MAINTENANCE MOE
2221 010344 104420 MRCK          ;CHECK MR REG
2222 010346 022701 22701        ;TO EQUAL 22701
2223 010350 104424 MRINT          ;INIT MR REG (CLEAR MRSP)
2224 010352 104430 MRIND          ;CLOCK INDEX PULSE IN RSMR
2225 010354 012777 000003 170526 MOV          #3,RSDA ;DO A SEARCH FOR SECTOR 3 OR 41
2226 010362 022777 000000 170536 CMP          #0,RSDT ;INTERLEAVED?
2227 010370 001403 BEQ          4$ ;NO SECTOR 3
2228 010372 012777 000041 170510 MOV          #41,RSDA ;YES SECTOR 41
2229 010400 012777 000031 170472 4$: MOV          #31,RSCSI ;LOAD SEARCH COMMAND (M7759)
2230 010406 104426 DSK          ;CHECK RSDS
2231 010410 030400 30400        ;TO EQUAL 30400
2232 010412 104000 HLT          ;PIP SHOULD BE SET AND DRY SHOULD
2233 : ;BE 0 FOR A DRIVE SEARCH CMD
2234 010414 012737 010643 001204 1$: MOV          #10643,REPT ;STEP THROUGH 3 SECTORS
2235 010422 104422 MRCLK          ;CLOCK MR
2236 010424 104426 DSK          ;RSDS SHOULD NOT
2237 010426 030400 30400        ;CHANGE TILL CLOCKING IS COMPLETED
2238 010430 104000 HLT          ;TO REACH SECTOR 3
2239 010432 005337 001204 DEC          REPT ;KEEP CLOCKING TILL
2240 010436 001371 BNE          1$ ;SECTOR 3 HAS BEEN REACHED
2241 : ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2242 010440 104446 MCLKI          ;CLOCK MR REG
2243 010442 104426 DSK          ;CHECK FOR "ATA" AND "DRY"
2244 010444 110600 110600        ;TO BE SET IN RSDS FOR
2245 010446 104000 HLT          ;SEARCH FUNCTION SHOULD BE COMPLETED
2246 010450 022777 104230 170422 CMP          #104230,RSCSI ;SET RCSI
2247 010456 001401 BEQ          2$ ;SC IN RCSI SHOULD SET BECAUSE OF
2248 010460 104140 HLT          !DS!AS ;COMPLETED SEARCH FUNCTION
2249 010462 013777 001164 170426 2$: MOV          UNITSV,RSAS ;CLEAR ATA
2250 010470 005777 170422 TST          RSAS ;DID ATA CLEAR BY WRITING INTO IT?
2251 010474 001401 BEQ          3$ ;YES
2252 010476 104140 HLT          !DS!AS ;RSAS SHOULD=0
2253 010500 022777 004230 170372 3$: CMP          #4230,RSCSI ;DID SC CLEAR BY CLEARING
2254 010506 001401 BEQ          +4 ;"ATA" YES
2255 010510 104140 HLT          !DS!AS ;NO

```

```

2256 ;*****
2257 ;TEST 46          BLOCK SEARCH TEST 2
2258 ;*****
2259 010512 104400  †TST46: SCOPE
2260
2261 ;MODULE TESTED: M7759, M7754, M7771, M7770
2262 ;THIS TEST INITIALIZES A BLOCK SEARCH FUNCTION FOR SECTOR 0, WHEN THE DRIVE
2263 ;IS CURRENTLY AT THE DESIRED SECTOR. THE BLOCK SEARCH FUNCTION
2264 ;SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION
2265 ;AND REACHES THE BEGINNING OF THE DESIRED SECTOR.
2266
2267 010514 104414  MRSRC: CLARK          ;CLEAR ALL REGISTERS
2268 010516 052737 000040 001170  BIS          #BITS,ONCEE ;SET CLOCK FLAG
2269 010524 104416  MRDMD          ;PUT DRIVE INTO MAINTENANCE MOE
2270 010526 104420  MRCK          ;CHECK MR REG
2271 010530 022701  22701        ;TO EQUAL 22701
2272 010532 104424  MRINT          ;INIT MR REG (CLEAR MRSF)
2273 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2274 010534 104430  MRIND
2275 010536 104420  MRCK          ;CHECK MR REG TO EQUAL
2276 010540 022701  22701        ;22701
2277 010542 104000  HLT
2278 ;STEP THRU RESYNC PERIOD
2279 010544 012737 001000 001204  MOV          #512,REPT
2280 010552 052737 000040 001170  BIS          #BITS,ONCEE ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2281 010560 104446  MRRT1: MCLK1        ;CLOCK MR REG
2282 010562 104420  MRCK          ;CHECK FOR
2283 010564 032711  32711        ;CORRECT DATA
2284 010566 104000  HLT          ;MR = BAD GOOD = CORRECT DATA
2285 010570 104450  MCLK0        ;CLOCK MR REG
2286 010572 104420  MRCK          ;CHECK FOR
2287 010574 022701  22701        ;CORRECT DATA
2288 010576 104000  HLT          ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2289 010600 005337 001204  DEC          REPT ;FINISH LOOPING
2290 010604 001365  BNE          MRRT1 ;THROUGH RESYNC PERIOD
2291 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
2292 010606 104446  MCLK1        ;CLOCK MR REG
2293 010610 104420  MRCK          ;MR SHOULD
2294 010612 032311  32311        ;EQUALS 32311
2295 010614 104000  HLT          ;MR=BAD GOOD=CORRECT ANS
2296 010616 104450  MCLK0        ;CLOCK MR REG
2297 010620 104420  MRCK          ;CHECK MR
2298 010622 022301  22301        ;TO EQUAL 22301
2299 010624 104000  HLT          ;MR=BAD GOOD=CORRECT ANS
2300 010626 012737 000100 001204  MOV          #100,REPT ;STEP INTO SECTOR 0
2301 010634 104422  25: MRCLK        ;CLOCK MR REG
2302 010636 005337 001204  DEC          REPT ;DO 100 TIMES
2303 010642 001374  BNE          25 ;DONE YET? NO BR
2304 010644 012777 000031 170226  45: MOV          #31,RSCS1 ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
2305 010652 104426  DSCK          ;CHECK RSDS
2306 010654 030400  30400        ;TO EQUAL 30400
2307 010656 104000  HLT          ;PIP SHOULD BE SET AND DRY SHOULD
2308 ;BE 0 FOR A DRIVE SEARCH CMD
2309 010660 012737 021506 001204  MOV          #21506,REPT ;STEP 3 SECTORS BEYOND SECTOR 0

```

```

2310 010666 104422          ;S:  MRCLK          ;CLOCK MR
2311 010670 104426          DSK          ;RSDS SHOULD NOT
2312 010672 030400          30400        ;CHANGE TILL CLOCKING IS COMPLETED
2313 010674 104000          HLT          ;TO REACH SECTOR 3
2314 010676 005337 001204  DEC      REPT    ;KEEP CLOCKING TILL
2315 010702 001371          BNE      1$     ;SECTOR 3 HAS BEEN REACHED
2316          ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2317 010704 104430          MRIND
2318 010706 104420          MRCK          ;CHECK MR REG TO EQUAL
2319 010710 022701          22701        ;22701
2320 010712 104000          HLT
2321
2322          ;STEP THRU RESYNC PERIOD
2323
2324 010714 012737 001000 001204  MOV      #512.,REPT
2325 010722 052737 000040 001170  BIS      #BITS,ONCEE
2326 010730 104446          MRWR1: MCLK1
2327 010732 104420          MRCK
2328 010734 032711          32711
2329 010736 104000          HLT
2330 010740 104450          MCLK0
2331 010742 104420          MRCK
2332 010744 022701          22701
2333 010746 104000          HLT
2334 010750 005337 001204  DEC      REPT
2335 010754 001365          BNE      MRWR1
2336
2337          ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2338          ;SP=0 EQUALS SECTOR PULSE
2339 010756 104446          MCLK1
2340 010760 104420          MRCK
2341 010762 032311          32311
2342 010764 104000          HLT
2343 010766 104450          MCLK0
2344 010770 104420          MRCK
2345 010772 022301          22301
2346 010774 104000          HLT
2347
2348          ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2349 010776 104446          MCLK1
2350 011000 104426          DSK
2351 011002 110600          110600
2352 011004 104000          HLT
2353 011006 022777 104230 170064  CMP      #104230,RS1
2354 011014 001401          BEQ
2355 011016 104140          HLT      !DS!AS
2356 011020 013777 001164 170070 2$:  MOV      UNITSV,RSAS
2357 011026 005777 170064          TST     RSAS
2358 011032 001401          BEQ      3$
2359 011034 104140          HLT     !DS!AS
2360 011036 022777 004230 170034 3$:  CMP      #4230,RS1
2361 011044 001401          BEQ
2362 011046 104140          HLT     !DS!AS ;NO

```

```

;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
;CLOCK MR REG
;CHECK FOR
;CORRECT DATA
;MR = BAD GOOD = CORRECT DATA
;CLOCK MR REG
;CHECK FOR
;CORRECT DATA
;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
;FINISH LOOPING
;THROUGH RESYNC PERIOD
;CLOCK MR REG
;MR SHOULD
;EQUAL 32311
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR
;TO EQUAL 22301
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK FOR "ATA" AND "DRY"
;TO BE SET IN RSDS FOR
;SEARCH FUNCTION SHOULD BE COMPLETED
;SET RSCS1
;SC IN RSCS1 SHOULD SET BECAUSE OF
;COMPLETED SEARCH FUNCTION
;CLEAR ATA
;DID ATA CLEAR BY WRITING INTO IT?
;YES
;RSAS SHOULD=0
;DID SC CLEAR BY CLEARING
;"ATA" YES

```

```

2363 :*****
2364 :TEST 47 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
2365 :*****
2366 011050 104400 TST47: SCOPE
2367
2368 :MODULE TESTED M7759, M7755, M7770
2369 :RMR ERROR IS CAUSED BY WRITTING INTO RSCS1 WHILE DOING A BLOCK SEARCH FUNCTION
2370 :CHECK RMR DECODER, E12, M7755, IF THIS TEST FAILS
2371
2372 011052 104414 RMRC1: CLRDK :CLEAR ALL DRIVE REGISTERS
2373 011054 042737 200040 C01170 BIC #BITS,ONCEE :CLEAR CLK CNT FLAG
2374 011062 104416 MRDMD :PUT DRIVE INTO MAINT MODE
2375 011064 104420 MRCK :CHECK MR REG TO
2376 011066 022701 22701 :EQUAL 22701
2377 011070 104424 MRINT :INIT MAINT MODE (CLEAR MRSP)
2378 011072 012777 000001 170010 MOV #1, @RSDA :LOAD RSDA
2379 011100 012777 000031 167772 MOV #31, @RSCS1 :LOAD BLOCK SEARCH FUNCTION
2380 011106 104426 DSK :CHECK RSDS
2381 011110 030400 30400 :TO EQUAL 30400
2382 011112 104000 HLT :DRY IN RSDS SHOULD BE
2383 :CLEARED FOR DRIVE WAS
2384 :ISSURED A BLOCK SEARCH FUNCTION
2385 :RSDS=BAD GOOD=CORRECT ANS
2386 011114 012777 000011 167756 MOV #11, @RSCS1 :LOAD A CLEAR FUNCTION
2387 :THIS SHOULD CAUSE AN RMR
2388 :ERROR FOR DRIVE WAS BUSY
2389 :WHEN CLEAR COMMAND WAS GIVEN
2390 011122 017700 167766 MOV @RSER, BAD :GET RSER REG
2391 011126 012701 000004 MOV #4, GOOD :GET CORRECT ANS
2392 011132 020100 CMP GOOD, BAD :DID RMR SET IN RSER?
2393 011134 001410 BEQ 1$ :YES
2394 011136 104402 022173 TYPE , TRMR :ASCIZ "RSCS1"
2395 011142 104402 011146 TYPE ,.+2 :RSER=BAD GOOD=CORRECT ANS
2396 011154 104000 HLT :CHECK RSDS TO
2397 011156 104426 1$: DSK :EQUAL 150600
2398 011160 150600 HLT :RSDS=BAD GOOD=CORRECT ANS
2399 011162 104000 HLT :DID CORRECT BITS SET IN RSCS1
2400 011164 022777 104230 167706 CMP #104230, @RSCS1 :YES
2401 011172 001401 BEQ 2$ :RSCS1 SHOULD=104230
2402 011174 104040 HLT !DS :RSDS SHOULD=150600
2403 :RSER SHOULD=4
2404 :DID CLR CLEAR RSDA
2405 011176 022777 000001 167704 2$: CMP #1, @RSDA :NO
2406 011204 001401 BEQ 4$ :RSDA SHOULD=1
2407 011206 104004 HLT !DA :CLEAR ALL REGISTERS
2408 011210 104414 4$: CLRDK :RSER SHOULD CLEAR
2409 011212 005777 167676 TST @RSER :RSER OK
2410 011216 001401 BEQ 3$ :RSER SHOULD=0 FOR THE
2411 011220 104040 HLT !DS :CLEAR BIT WAS LOADED IN RSCS2
2412 :RSCS1 SHOULD=4200 FOR THE
2413 011222 022777 004200 167650 3$: CMP #4200, @RSCS1 :CLEAR BIT WAS LOADED IN RSCS2
2414 011230 001401 BEQ .+4 :RSCS1 SHOULD=4200
2415 011232 104040 HLT !DS

```

2416
2417
2418
2419 011234 104400
2420
2421
2422
2423
2424 011236 104414
2425 011240 104416
2426 011242 104420
2427 011244 022701
2428 011246 104424
2429 011250 012777 000001 167632
2430 011256 012777 000031 167614
2431 011264 104426
2432 011266 030400
2433 011270 104000
2434
2435
2436
2437 011272 005077 167612
2438
2439
2440
2441 011276 017700 167612
2442 011302 012701 000004
2443 011306 020100
2444 011310 001410
2445 011312 104402 022173
2446 011316 104402 011322
2447 011330 104000
2448 011332 104426 15:
2449 011334 150600
2450 011336 104000
2451 011340 022777 104230 167532
2452 011346 001401
2453 011350 104040
2454
2455
2456 011352 022777 000001 167530 25:
2457 011360 001401
2458 011362 104004
2459 011364 104414 45:
2460 011366 005777 167522
2461 011372 001401
2462 011374 104040
2463
2464 011376 022777 004200 167474 35:
2465 011404 001401
2466 011406 104040
2467

```
*****
:TEST 50 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
*****
TST50: SCOPE

:MODULE TESTED M7755 M7759 M7770
:RMR ERROR IS CAUSED BY WRITTING INTO RSDA WHILE DOING A BLOCK SEARCH FUNCTION

RMR02: CLRDK ;CLEAR ALL DRIVE REGISTERS
MRDMD ;PUT DRIVE INTO MAINT MODE
MRCK ;CHECK MR REG TO
22701 ;EQUAL 22701
MRINT ;INIT MAINT MODE (CLEAR MRSP)
MOV #1,RSDA ;LOAD RSDA
MOV #31,RSCS1 ;LOAD BLOCK SEARCH FUNCTION
DSCK ;CHECK RSDS
30400 ;TO EQUAL 30400
HLT ;DRY IN RSDS SHOULD BE
;CLEARED FOR DRIVE WAS
;ISSURED A BLOCK SEARCH FUNCTION
;RSDS=BAD GOOD=CORRECT ANS
CLR RSDA ;MODIFY RSDA
;THIS SHOULD CAUSE AN RMR
;ERROR FOR DRIVE WAS BUSY
;WHEN COMMAND WAS GIVEN
MOV RRSER,BAD ;GET RSER REG
MOV #4,GOOD ;GET CORRECT ANS
CMP GOOD,BAD ;DID RMR SET IN RSER?
BEQ 15 ;YES
TYPE ,TRMR
TYPE ,.+2
;ASCIZ "RSDA"
HLT ;RSDA SHOULD=1
DSCK ;CHECK RSDS TO
150600 ;EQUAL 150600
HLT ;RSDS=BAD GOOD=CORRECT ANS
CMP #104230,RSCS1 ;DID CORRECT BITS SET IN RSCS1
BEQ 25 ;YES
HLT ;RSCS1 SHOULD=104230
;RSDS SHOULD=150600
;RSER SHOULD=4
25: CMP #1,RSDA ;DID CLR CLEAR RSDA
BEQ 45 ;NO
HLT ;RSDA SHOULD=1
;CLEAR ALL REGISTERS
45: CLRDK ;RSDA SHOULD CLEAR
TST RRSER ;RSER OK
BEQ 35 ;RSER SHOULD=0 FOR THE
HLT ;CLEAR BIT WAS LOADED IN RSCS2
;RSCS1 SHOULD=4200 FOR THE
35: CMP #4200,RSCS1 ;CLEAR BIT WAS LOADED IN RSCS2
BEQ .+4 ;RSCS1 SHOULD=4200
HLT ;DS
```

K05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST51

RS11-REG3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 63
DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)

```

2468 :*****
2469 :TEST 51 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
2470 :*****
2471 011410 104400 TST51: SCOPE
2472
2473 :MODULE TESTED M7759, M7755, M7770
2474 :RMR ERROR IS CAUSED BY WRITTING INTO RSER WHILE DOING A BLOCK SEARCH FUNCTION
2475 :CHECK RMR DECODER, E12-M7755, IF THIS TEST FAILS.
2476
2477 011412 104414 RMRC3: CLRDK :CLEAR ALL DRIVE REGISTERS
2478 011414 042737 000040 001170 BIC #BITS,ONCEE :CLEAR CLOCK COUNT FLAG
2479 011422 104416 MRDMD :PUT DRIVE INTO MAINT MODE
2480 011424 104420 MRCK :CHECK MR REG TO
2481 011426 022701 22701 :EQUAL 22701
2482 011430 104424 MRINT :INIT MAINT MODE (CLEAR MRSP)
2483 011432 012777 000001 167450 MOV #1,RSOA :LOAD RSOA
2484 011440 012777 000031 167432 MOV #31,RSCSI :LOAD BLOCK SEARCH FUNCTION
2485 011446 104426 DSCK :CHECK RSDS
2486 011450 030400 30400 :TO EQUAL 30400
2487 011452 104000 HLT :DRY IN RSDS SHOULD BE
2488 :CLEARED FOR DRIVE WAS
2489 :ISSURED A BLOCK SEARCH FUNCTION
2490 :RSDS=BAD GOOD=CORRECT ANS
2491 011454 012777 177777 167432 MOV #-1,RSER :MODIFY RSER
2492 :THIS SHOULD CAUSE AN RMR
2493 :ERROR FOR DRIVE WAS BUSY
2494 :WHEN COMMAND WAS GIVEN
2495 011462 017700 167426 MOV RSER,BAD :GET RSER REG
2496 011466 012701 000004 MOV #4,GOOD :GET CORRECT ANS
2497 011472 020100 CMP GOOD,BAD :DID RMR SET IN RSER?
2498 011474 001410 BEQ 1$ :YES
2499 011476 104402 022173 TYPE ,TRMR :ASCIZ "RSER"
2500 011502 104402 011506 TYPE ,+2 :RSER=BAD GOOD=CORRECT ANS
2501 011514 104000 HLT :CHECK RSDS TO
2502 011516 104426 1$: DSCK :EQUAL 150600
2503 011520 150600 150600 :RSDS=BAD GOOD=CORRECT ANS
2504 011522 104000 HLT :DID CORRECT BITS SET IN RSCS1
2505 011524 022777 104230 167346 CMP #104230,RSCSI :YES
2506 011532 001401 BEQ 4$ :RSCS1 SHOULD=104230
2507 011534 104040 HLT !DS :RSDS SHOULD=150600
2508 :RSER SHOULD=4
2509 :CLEAR ALL REGISTERS
2510 011536 104414 4$: CLRDK :CLEAR ALL REGISTERS
2511 011540 005777 167350 TST RSER :RSER SHOULD CLEAR
2512 011544 001401 BEQ 3$ :RSER OK
2513 011546 104040 HLT !DS :RSER SHOULD=0 FOR THE
2514 :CLEAR BIT WAS LOADED IN RSCS2
2515 011550 022777 004200 167322 3$: CMP #4200,RSCSI :RSCS1 SHOULD=4200 FOR THE
2516 011556 001401 BEQ +4 :CLEAR BIT WAS LOADER IN RSCS2
2517 011560 104040 HLT !DS :RSCS1 SHOULD=4200

```

L05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST52

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 64
DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)

```

2518 ;*****
2519 ;TEST 52 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
2520 ;*****
2521 011562 104400 †TST52: SCOPE
2522
2523 ;MODULE TESTED: M7759, M7755, M7770
2524 ;RMR ERROR SHOULD NOT SET BY WRITTING INTO RSAS WHILE DOING A BLOCK SEARCH FUNCTION
2525 ;IF TEST FAILS, CHECK RMR DECODER E12-M7755.
2526
2527 011564 104414 RMR04: CLR0K :CLEAR ALL DRIVE REGISTERS
2528 011566 104416 MR0MD :PUT DRIVE INTO MAINT MODE
2529 011570 104420 MRCK :CHECK MR REG TO
2530 011572 022701 22701 :EQUAL 22701
2531 011574 104424 MRINT :INIT MAINT MODE (CLEAR MRSP)
2532 011576 012777 000001 167304 MOV #1, @RSDA :LOAD RSDA
2533 011604 012777 000031 167266 MOV #31, @RSCS1 :LOAD BLOCK SEARCH FUNCTION
2534 011612 104426 DSCK :CHECK RSDS
2535 011614 030400 30400 :TO EQUAL 30400
2536 011616 104000 HLT :DRY IN RSDS SHOULD BE
2537 :CLEARED FOR DRIVE WAS
2538 :ISSURED A BLOCK SEARCH FUNCTION
2539 :RSDS=BAD GOOD=CORRECT ANS
2540 011620 005077 167272 CLR @RSAS :WRITE INTO ATTENTION SUMMARY REGISTER
2541 :SHOULD BE NO RMR ERROR BECAUSE
2542 :WRITING RSAS IN ALLOWED ANYTIME.
2543 011624 017700 167264 MOV @RSER, BAD :GET RSER REG
2544 011630 012701 000000 MOV #0, GOOD :GET CORRECT ANS
2545 011634 020100 CMP GOOD, BAD :DID RMR SET IN RSER?
2546 011636 001435 BEQ 1$ :NO
2547 011640 104402 011644 TYPE ,.+2 :.ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
2548 011730 104000 HLT :RSDS=BAD GOOD=CORRECT ANS
2549 011732 104426 1$: DSCK :CHECK RSDS TO
2550 011734 030400 30400 :EQUAL 30400
2551 011736 104000 HLT :RSDS=BAD GOOD=CORRECT ANS
2552 011740 022777 004231 167132 CMP #4231, @RSCS1 :DID CORRECT BITS SET IN RSCS1
2553 011746 001401 BEQ 4$ :YES
2554 011750 104040 HLT !DS :RSCS1 SHOULD=4231
2555 :RSDS SHOULD=30400
2556 :RSER SHOULD=0
2557 011752 104414 4$: CLR0K :CLEAR ALL REGISTEREDS
2558 011754 005777 167134 TST @RSER :RSER SHOULD CLEAR
2559 011760 001401 BEQ 3$ :RSER OK
2560 011762 104040 HLT !DS :RSER SHOULD=0 FOR THE
2561 :CLEAR BIT WAS LOADED IN RSCS2
2562 011764 022777 004200 167106 3$: CMP #4200, @RSCS1 :RSCS1 SHOULD=4200 FOR THE
2563 011772 001401 BEQ .+4 :CLEAR BIT WAS LOADER IN RSCS2
2564 011774 104040 HLT !DS :RSCS1 SHOULD=4200

```


M05

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST53

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC
DRIVE SELECT TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 65

```

2565 ;*****
2566 ;TEST 53 DRIVE SELECT TEST
2567 ;*****
2568 011776 104400 †TST53: SCOPE
2569
2570 ;MODULE TESTED: M7755
2571 ;THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES.
2572 ;THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND TRIES TO LOAD ITS
2573 ;REGISTER WITH ALL ZEROS. THIS SHOULD CAUSE "NED" TO
2574 ;SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS
2575 ;ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.
2576 ;CHECK UNIT NO. COMPARATOR, E19-M7755 IF TEST FAILS
2577
2578 012000 104414 MRDSEL: CLRDK ;CLEAR ALL REGISTERS
2579 012002 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2580 012004 104420 MRCK ;CHECK MAINT REG
2581 012006 022701 22701 ;TO EQUAL 22701
2582 012010 104424 MRINT ;INITIALIZE MAINT MODE (CLEAR MRSP)
2583
2584 012012 012777 177777 167070 MOV #-1,ARSDA ;LOAD DISK ADDR REG OF DRIVE UNDER TEST
2585
2586 ;SEARCH FOR NON EXISTENT DRIVES
2587
2588 012020 012737 000401 001216 MOV #401,WORK
2589 012026 005001 CLR GOOD
2590 012030 010177 167046 1S: MOV GOOD,ARSCS2 ;LOAD UNIT NO
2591 012034 005777 167054 TST ARSER ;IS THIS A NED?
2592 012040 032777 010000 167034 BIT #BIT12,ARSCS2 ;IS THIS A NED?
2593 012046 001005 BNE 2S ;FOUND NED
2594 012050 005201 INC GOOD ;UPDATE UNIT NUMBER
2595 012052 006137 001216 ROL WORK ;KEEP LOOKING FOR NED
2596 012056 103460 BCS NEDDON ;COULD NOT FIND ANY NON EXISTENT DRIVES
2597 012060 000763 BR 1S ;LOOK FOR NED
2598 012062 012777 004000 167010 2S: MOV #4000,ARSCS1 ;CLEAR NED
2599 012070 010137 001222 MOV GOOD,WORK1 ;SAVE NED NUMBER
2600 012074 010177 167002 MOV GOOD,ARSCS2 ;LOAD UNIT # OF NED INTO RSCS2
2601 012100 005077 167004 CLR ARSDA ;WRITE INTO A NON EXISTENT DRIVE REG
2602 ;THIS SHOULD CAUSE NED TO
2603 ;SET IN RSCS2
2604 012104 017700 166772 MOV ARSCS2,BAD ;GET RSCS2
2605 012110 052701 010100 BIS #10100,GOOD ;PUT CORRECT ANS IN GOOD
2606 ;BY SETTING NED AND IR
2607 012114 020100 CMP GOOD,BAD ;IS RSCS2 CORRECT?
2608 012116 001401 BEQ .+4 ;YES
2609 012120 104000 HLT ;RSCS2=BAD GOOD=CORRECT ANS
2610
2611 012122 022777 160200 166750 CMP #160200,ARSCS1 ;IS CS1 CORRECT
2612 012130 001401 BEQ .+4 ;YES
2613 012132 104004 HLT !DA ;TRE SHOULD BE SET IN CS1 BECAUSE
2614 ;OF NED ERROR IN RSCS2
2615 ;RSCS1 SHOULD=160200

```

2616	012134	005777	166756		TST	DRSAS		: DID ANY ATTENTION BITS SET?
2617	012140	001401			BEQ	.+4		: NO
2618	012142	104100			HLT	!AS		: NO ATTENTION BITS SHOULD BE SET
2619	012144	112777	000100	166762	MOV	#100,DRSCS1B		: CLEAR TRE
2620	012152	032777	010000	166722	BIT	#NED,DRSCS2		: DID NED CLEAR
2621	012160	001401			BEQ	.+4		: YES
2622	012162	104040			HLT	!DS		: NED DID NOT CLEAR IN RSCS2
2623								: BY CLEARING TRE BIT IN RSCS1
2624	012164	013777	001162	166710	MOV	UNNUM,DRSCS2		: LOAD CORRECT UNIT NUMBER
2625	012172	022777	177777	166710	CMP	#-1,DRSDA		: DID RSDA GET MODIFIED
2626								: WHILE WRITING INTO A NON
2627								: EXISTENT DRIVE?
2628	012200	001443			BEQ	NNDD		: NO
2629	012202	104004			HLT	!DA		: RSDA SHOULD= -1
2630	012204	013700	001222		MOV	WORK1,BAD		: IT GOT MODIFIED WHILE WRITING
2631	012210	013701	001162		MOV	UNNUM,GOOD		: INTO A -ED
2632	012214	104000			HLT			: GOOD=DRIVE UNDER TEST
2633	012216	000434			BR	NNDD		: BAD=NON EXISTENT DRIVE THAT WAS
2634								: IN RSCS2 WHEN RSDA GOT MODIFIED
2635	012220	032737	010000	001170	NEEDON: BIT	#BIT12,ONCEE		: WAS THIS TYPED BEFORE?
2636	012226	001030			BNE	NNDD		: YES
2637	012230	104402	012234		TYPE	.+2		: .ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2638	012302	052737	010000	001170	BIS	#BIT12,ONCEE		: SET TYPED MESSAGE FLAG
2639	012310				NNDD:			

2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693

012310 104400

012312 012737 001602 001144
012320 104414
012322 012737 000040 001170
012330 104430
012332 104420
012334 022701
012336 104424

012340 012702 030114
012344 005022
012346 012722 177777
012352 012722 177777
012354 012722 177777
012356 006103
012360 103402
012362 010322
012364 000774
012366 012702 000056

012372 012704 146314
012376 010422
012400 005303
012402 001375

012404 012777 030114 166474
012412 012777 177700 166464
012420 012777 000061 166452
012426 104454

012430 104220
012432 104456

```
*****
:TEST 54 MAINTENANCE MODE WRITE TEST
*****
TST54: SCOPE

:MODULE TESTED: M7771, M7753, M7751
:THIS IS AN RSQ3 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR
:WRITE TEST. WE ARE TESTING THE COMPLETE DATA PATH FOR A DATA
:TRANSFER TO THE DISK. MILLAR ENCODED DATA TO BOTH SURFACES IS
:CHECKED ALONG WITH CORRECT GENERATION OF THE CRC WORD AT THE END
:OF THE SECTOR. INDEX PULSES, RESYNC, TIMING PREAMBLE, AND SECTOR
:PULSES ARE ALSO CHECKED.

MRWRT: MOV #1602,FLAG2 ;SET TEST FLAG
        CLR DRK ;CLEAR DRIVE REGISTERS
        MOV #40,ONCEE ;SETUP TEST FLAGS
        PRIND ;SEND INDEX PULSE TO MR REG
        MRCK ;CHECK MR REG
        22701 ;TO EQUAL 22701
        MRINT ;INIT MAINT MODE (CLEAR MRSP)
        ;BY SENDING 2 CLOCK PULSES

:FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
:DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
: :A WORD OF ALL 1'S
: :FLOATING 1'S PATTERN (16 WORDS)
: :A PATTERN OF 146314 (46 WORDS)
:

        MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
        CLR (R2)+ ;CLEAR 1ST LOCATION
        MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
        CLR R3 ;CLEAR WORK LOC TO GENERATE
        SEC ;A PATTERN OF FLOATING ONES
        1$: ROL R3 ;GET PATTERN
        BCS 2$ ;DONE GET OUT
        MOV R3,(R2)+ ;FILL BUFFER
        BR 1$ ;CONT
        2$: MOV #46.,R3 ;FILL REMAINING PORTION OF
        3$: MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
        MOV R4,(R2)+ ;LOAD BUFFER
        DEC R3 ;DONE YET?
        BNE 3$ ;NO

:SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) TO SECTOR 0

        MOV #INBUF,DRSBA ;LOAD BUS ADDR REG
        MOV #177700,DRSBC ;LOAD WORD COUNT REG
        MOV #61,DRSCS1 ;LOAD WRITE COMMAND
        GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
        ;TO CLEAR OUT COUNTERS AND REGISTERS
        ;THAT OTHERWISE COULD NOT BE CLEARED.
        HLT !MR ;COULD NOT SET SECTOR PULSE (0)
        SPASS ;CLOCK MR REG SP = 1
```

```

2694                                     ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2695 012434 104430 MRIND
2696 012436 104420 MRCK                                     ;CHECK MR REG TO EQUAL
2697 012440 020501 20501                               ;20501 FOR A
2698 012442 104060 HLT                               ;WRITE COMD HAS BEEN ISSUED
2699
2700                                     ;STEP THRU RESYNC PERIOD
2701
2702 012444 012737 001000 001204 MOV #512.,REPT
2703 012452 052737 000040 001170 BIS #BITS,ONCEE
2704 012460 104446 MRWRT1: MCLK1                                     ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
2705 012462 104420 MRCK                                     ;CLOCK MR REG
2706 012464 030511 30511                                     ;CHECK FOR
2707 012466 104000 HLT                                     ;CORRECT DATA
2708 012470 104450 MCLK0                                     ;MR = BAD GOOD = CORRECT DATA
2709 012472 104420 MRCK                                     ;CLOCK MR REG
2710 012474 020501 20501                                     ;CHECK FOR
2711 012476 104000 HLT                                     ;CORRECT DATA
2712 012500 005337 001204 DEC REPT                               ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2713 012504 001365 BNE MRWRT1                               ;FINISH LOOPING
2714                                     ;THROUGH RESYNC PERIOD
2715
2716                                     ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2717 012506 104446 MCLK1                                     ;SP=0 EQUALS SECTOR PULSE
2718 012510 104420 MRCK                                     ;CLOCK MR REG
2719 012512 030111 30111                                     ;MR SHOULD
2720 012514 104000 HLT                                     ;EQUAL 30111
2721 012516 104450 MCLK0                                     ;MR=BAD GOOD=CORRECT ANS
2722 012520 104420 MRCK                                     ;CLOCK MR REG
2723 012522 020101 20101                                     ;CHECK MR
2724 012524 104000 HLT                                     ;TO EQUAL 20101
2725                                     ;MR=BAD GOOD=CORRECT ANS
2726
2727                                     ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE
2728 012526 012737 000077 001204 MRWRT2: MOV #63.,REPT
2729 012534 104446 MRWRT2: MCLK1                                     ;CLOCK MR REG
2730 012536 104420 MRCK                                     ;CHECK MR REG
2731 012540 031511 31511                                     ;TO EQUAL 31511
2732 012542 104000 HLT                                     ;MR=BAD GOOD=CORRECT ANS
2733 012544 104450 MCLK0                                     ;CLOCK MR REG
2734 012546 104420 MRCK                                     ;CHECK MR REG
2735 012550 021501 21501                                     ;TO EQUAL 21501
2736 012552 104000 HLT                                     ;MR=BAD GOOD=CORRECT ANS
2737 012554 005337 001204 DEC REPT                               ;DONE YET
2738 012560 001365 BNE MRWRT2                               ;NO LOOP

```

```

2739                                     ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
2740
2741 012562 104446 MCLK1 ;CLOCK MR REG
2742 012564 104420 MRCK ;CHECK MR REG
2743 012566 131511 131511 ;TO EQUAL 131511
2744 012570 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2745 012572 104450 MCLK0 ;CLOCK MR REG
2746 012574 104420 MRCK ;MR REG SHOULD
2747 012576 025501 25501 ;EQUAL 25501
2748 012600 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2749 012602 104446 MCLK1 ;CLOCK MR REG
2750 012604 104420 MRCK ;MR SHOULD EQUAL
2751 012606 135511 135511 ;35511
2752 012610 104000 HLT
2753                                     ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
2754 012612 012737 000010 001204 MOV #10,REPT
2755 012620 104450 MRWRT3: MCLK0 ;CLOCK MR REG
2756 012622 104420 MRCK ;CHECK MR REG
2757 012624 025501 25501 ;TO EQUAL 25501
2758 012626 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2759 012630 104446 MCLK1 ;CLOCK MR REG
2760 012632 104420 MRCK ;CHECK MR REG
2761 012634 135511 135511 ;TO EQUAL 135511
2762 012636 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2763 012640 005337 001204 DEC REPT ;DONE YES?
2764 012644 001365 BNE MRWRT3 ;NO LOOP BACK
2765
2766                                     ;MOVE DATA WORD INTO RS03 SHIFT REGISTER (M7753)
2767
2768 012646 104450 MCLK0 ;CLOCK MR REG
2769 012650 104420 MRCK ;CHECK MR REG
2770 012652 021501 21501 ;TO EQUAL 21501
2771 012654 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2772
2773                                     ;ENCODE SYNC 1 (M7751)
2774
2775 012656 104446 MCLK1 ;CLOCK MR REG
2776 012660 104420 MRCK ;MR REG SHOULD
2777 012662 123511 123511 ;EQUAL 123511
2778 012664 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2779 012666 104450 MCLK0 ;CLOCK MR REG
2780 012670 104420 MRCK ;MR REG SHOULD NOW
2781 012672 033501 33501 ;EQUAL 33501
2782 012674 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2783 012676 012705 030114 MOV #INBUF,R5 ;GET STARTING ADDR FOR DATA BUFFER
2784 012702 011504 MOV (R5),R4 ;GET DATA

```

E06

MAINDEC-11-DZRSE-C
DZRSEC.P11

TST54

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE MODE WRITE TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 70

2785	012704	012737	002156	001216		MOV	#1134.,WORK	:DOING A 1 SECTOR TRANSFER 63 WORDS
2786								:18 BITS PER WORD-CLOCK LOOPS
2787								:TAKE CARE OF 1 BIT AT A TIME
2788								:63 TIMES 18 EQUALS 1134 LOOPS
2789								:TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY)
2790	012712	052737	000100	001170	1\$:	BIS	#BIT6,ONCEE	:SET 1ST TRANSFER WORD FLAG
2791	012720	104432				XBIT		:GET 1 BIT OF DATA
2792	012722	104434				CLKD1		:SET MCLK IN RSMR
2793								:AND CALCULATE MR REG
2794								:FOR CORRECT DATA (MWOB)
2795	012724	104000				HLT		:MR REG NOT CORRECT
2796	012726	104436				CLKD0		:CLEAR MCLK TO
2797								:COMPLETE TRANSFER OF THIS BIT
2798								:CALCULATE CORRECT ANS FOR
2799								:MR REG (MWOB)
2800	012730	104000				HLT		:MR=BAD GOOD=CORRECT ANS
2801	012732	032737	000200	001170		BIT	#BIT7,ONCEE	:ON LAST WORD YET?
2802	012740	001015				BNE	2\$:YES
2803	012742	032737	000400	001170		BIT	#BIT8,ONCEE	:ON CRC WORD YET?
2804	012750	001040				BNE	3\$:YES
2805	012752	005337	001216			DEC	WORK	:DONE WITH 63 WORDS?
2806	012756	001360				BNE	1\$:NO
2807								
2808	012760	052737	000200	001170		BIS	#BIT7,ONCEE	:SET LAST WORD FLAG
2809	012766	012737	000023	001216	2\$:	MOV	#19.,WORK	:SET UP TO TRANSFER LAST WORD
2810	012774	005337	001216			DEC	WORK	:DONE YET?
2811	013000	001347				BNE	1\$:NO
2812	013002	052737	000400	001170		BIS	#BIT8,ONCEE	:SET TRANSFERRING CRC WORD
2813	013010	042737	000200	001170		BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
2814	013016	004737	024430			JSR	PC,GENCRC	:GENERATE CRC WORD
2815								:AND LEAVE IN "WORK"
2816	013022	012702	030114			MOV	#INBUF,R2	:GO TO END
2817	013026	062702	000200			ADD	#200,R2	:OF DATA BUFFER
2818	013032	013712	001216			MOV	WORK,R2	:LOAD CRC WORD
2819	013036	010205				MOV	R2,R5	:RESET POINTER FOR
2820	013040	162705	000002			SUB	#2,R5	:R5 FOR CRC WD
2821	013044	012737	000023	001216	3\$:	MOV	#19.,WORK	:SETUP TO XFER CRC
2822	013052	005337	001216			DEC	WORK	:DONE YET
2823	013056	001320				BNE	1\$:NO
2824								
2825								:EBL SHOULD NOW ASSERT
2826								
2827	013060	104446				MCLK1		:CLOCK MR REG TO STOP THROUGH
2828								:THE RS03 SECTOR DEAD BAND AREA
2829	013062	104420				MRCK		:CHECK MR REG
2830	013064	113511				113511		:TO EQUAL 113511
2831	013066	104000				HLT		:MR REG=BAD GOOD=CORRECT ANS

```

2832                :LOOP 17 TIMES
2833
2834 013070 012737 000017 001204      MOV      #17,REPT
2835 013076 104450      4S:      MCLKO
2836 013100 104420      MRCK      ;CLOCK MR REG
2837 013102 003501      3501     ;CHECK MR REG
2838 013104 104000      HLT      ;TO EQUAL 3501
2839 013106 104446      MCLK1    ;MR=BAD GOOD=CORRECT ANS
2840 013110 104420      MRCK      ;CLOCK MR REG
2841 013112 113511      113511  ;CHECK MR REG
2842 013114 104000      HLT      ;TO EQUAL 113511
2843 013116 005337 001204      DEC      ;MR=BAD GOOD=CORRECT ANS
2844 013122 001365      BNE      REPT  ;DONE LOOPING YET?
2845                4S
2846                ;FINISH UP
2847 013124 104450      MCLKO
2848 013126 104420      MRCK      ;CLOCK MR REG
2849 013130 003501      3501     ;CHECK MR REG
2850 013132 104000      HLT      ;TO EQUAL 3501
2851 013134 104446      MCLK1    ;MR REG=BAD GOOD=CORRECT ANS
2852 013136 104420      MRCK      ;CLOCK MR REG
2853 013140 111511      111511  ;CHECK MR REG
2854 013142 104000      HLT      ;TO EQUAL 111511
2855 013144 104450      MCLKO    ;MR=BAD GOOD=CORRECT ANS
2856 013146 104420      MRCK      ;CLOCK MR REG
2857 013150 001501      1501     ;CHECK MR REG
2858 013152 104000      HLT      ;TO EQUAL 1501
2859                ;MR=BAD GOOD=CORRECT ANS
2860                ;TRANSFER SHOULD NOW BE COMPLETE
2861
2862 013154 104446      MCLK1
2863 013156 104420      MRCK      ;CLOCK MR REG
2864 013160 012711      12711    ;CHECK MR
2865 013162 104000      HLT      ;REG TO
2866 013164 104450      MCLKO    ;EQUAL 12711
2867 013166 104420      MRCK      ;CLOCK MR REG
2868 013170 002701      2701     ;CHECK MR REG
2869 013172 104000      HLT      ;TO
2870                ;EQUAL 2701
2871                ;NOW TEST CONTROLLER
2872
2873 013174 005777 165700      TST      @RSCS1
2874 013200 100001      BPL      5S
2875 013202 104014      HLT      !DA!WC
2876 013204 005777 165674      5S:      TST      @RSCWC
2877 013210 001401      BEQ      .+4
2878 013212 104010      HLT      !WC
2879 013214 022777 000001 165666      CMP      #1,@RSDA
2880 013222 001401      BEQ      .+4
2881 013224 104004      HLT      !DA
2882 013226 032737 000002 001144      BIT      @BIT1,FLAG2
2883 013234 001002      BNE      .+6
2884 013236 000137 021450      JMP      @#MRVR2

```

```

;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 3501
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 113511
;MR=BAD GOOD=CORRECT ANS
;DONE LOOPING YET?
;NO

;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 3501
;MR REG=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 111511
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 1501
;MR=BAD GOOD=CORRECT ANS

;CLOCK MR REG
;CHECK MR
;REG TO
;EQUAL 12711
;CLOCK MR REG
;CHECK MR REG
;TO
;EQUAL 2701

;ANY ERRORS?
;NO
;YES
;DID WC GO TO 0
;YES
;WC SHOULD BE = TO 0
;DID RSDA INCREMENT TO A 1
;YES
;NO RSDA SHOULD=1
;IN MAINT VERIFY TEST?
;NO
;YES, GO TO VERIFY TEST

```

```

2885 :*****
2886 :TEST 55 MAINTENANCE READ TEST
2887 :*****
2888 013242 104400 TST55: SCOPE
2889
2890 ;MODULE TESTED: M7771, M7753, M7751
2891 ;THIS IS AN RS03 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR READ TIMING
2892 ;TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE DISK DECODING LOGIC
2893 ;TO CORE MEMORY. (THE PHASE LOCK LOOP IS NOT TESTED)
2894
2895 013244 104414 MRRD: CLRDK ;CLEAR DRIVE REGISTERS
2896 013246 052737 000040 001170 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
2897 013254 042737 047716 001170 BIC #47716,ONCEE ;CLEAR ALL OTHER FLAG BITS
2898 013262 104430 MRIND ;SEND INDEX PULSE TO MR REG
2899 013264 104420 MRCK ;CHECK MR REG
2900 013266 022701 22701 ;TO EQUAL 22701
2901 013270 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2902 ;BY SENDING 2 CLOCK PULSES
2903
2904 013272 005037 001144 CLR FLAG2 ;CLEAR FLAG TEST BITS
2905
2906 ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
2907 ;DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
2908 ; :A WORD OF ALL 1'S
2909 ; :FLOATING 1'S PATTERN (16 WORDS)
2910 ; :A PATTERN OF 146314 (46 WORDS)
2911
2912 013276 012702 030114 MOV #INBUF,R2 ;GET LOCATION OF INBUF
2913 013302 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
2914 013304 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
2915 013310 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
2916 013312 000261 SEC ;A PATTERN OF FLOATING ONES
2917 013314 006103 15: ROL R3 ;GET PATTERN
2918 013316 103402 BCS 25 ;DONE GET OUT
2919 013320 010322 MOV R3,(R2)+ ;FILL BUFFER
2920 013322 000774 BR 15 ;CONT
2921 013324 012703 000056 25: MOV #46,R3 ;FILL REMAINING PORTION OF
2922 013330 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
2923 013334 010422 35: MOV R4,(R2)+ ;LOAD BUFFER
2924 013336 005303 DEC R3 ;DONE YET
2925 013340 001375 BNE 35 ;NO
2926
2927 ;NOTE
2928 ;INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ"
2929 ;VIA THE MR08 BIT IN RSMR.
2930 ;OUTBUF IS WHERE THE DATA WORDS FROM THE
2931 ;MASSBUS ARE STORED.

```



```

2932 ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2933
2934 013342 012777 030714 165536 MOV #OUTBUF, @RSBA ;LOAD BUS ADDR REG
2935 013350 012777 177700 165526 MOV #177700, @RSWC ;LOAD WORD COUNT REG
2936 013356 012777 000071 165514 MOV #71, @RSC51 ;LOAD READ COMMAND
2937 013364 012702 000100 MOV #100, R2 ;CLEAR THE OUTBUF TABLE SO THAT
2938 013370 012703 030714 MOV #OUTBUF, R3 ;WHEN THE READ IS FINISHED, WE CAN
2939 013374 005023 45: CLR (R3)+ ;COMPARE WHAT WE GOT (OUTBUF)
2940 013376 005302 DEC R2 ;WITH WHAT WE EXPECTED (INBUF).
2941 013400 001375 BNE 45
2942 013402 104454 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
2943 ;TO CLEAR OUT COUNTERS AND REGISTERS
2944 ;THAT OTHERWISE COULD NOT BE CLEARED.
2945 013404 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
2946 013406 104456 SPASS ;CLOCK MR REG SP = 1
2947
2948 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2949 013410 104430 MRIND
2950 013412 104420 MRCK ;CHECK MR REG TO EQUAL
2951 013414 022601 22601 ;22601 FOR A
2952 013416 104000 HLT ;READ COMD
2953
2954 ;STEP THRU RESYNC PERIOD
2955
2956 013420 012737 001000 001204 MOV #512, REPT
2957 013426 052737 000040 001170 BIS #BITS, @NCEE ;TYPE OUT CLOCK COUNT
2958 013434 104446 MRRD1: MCLK1 ;CLOCK MR REG
2959 013436 104420 MRCK ;CHECK FOR
2960 013440 032611 32611 ;CORRECT DATA
2961 013442 104000 HLT ;MR=BAD GOOD=CORRECT DATA
2962 013444 104450 MCLK0 ;CLOCK MR REG
2963 013446 104420 MRCK ;CHECK FOR
2964 013450 022601 22601 ;CORRECT DATA
2965 013452 104000 HLT ;ERROR WHILE CLOCKING THROUGH RESYNC
2966 013454 005337 001204 DEC REPT ;FINISH LOOPING
2967 013460 001365 BNE MRRD1 ;THROUGH RESYNC PERIOD
2968
2969 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2970 ;SP=0 EQUALS SECTOR PULSE
2971 013462 104446 MCLK1 ;CLOCK MR REG
2972 013464 104420 MRCK ;MR SHOULD
2973 013466 032211 32211 ;EQUAL 32211
2974 013470 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2975 013472 104450 MCLK0 ;CLOCK MR REG
2976 013474 104420 MRCK ;CHECK MR
2977 013476 022201 22201 ;TO EQUAL 22201
2978 013500 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```

```

2979          ;PERFORM 71 MAINT CLOCK OPERATIONS--
2980
2981 013502 012737 000107 001204      MOV      #71.,REPT
2982 013510 104446      MRRD2:  MCLK1      ;CLOCK MR REG
2983 013512 104420      MRCK      ;CHECK MR REG
2984 013514 033611      33611    ;TO EQUAL 33611
2985 013516 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
2986 013520 104450      MCLK0    ;CLOCK MR REG
2987 013522 104420      MRCK      ;CHECK MR REG
2988 013524 023601      23601    ;TO EQUAL 23601
2989 013526 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
2990 013530 005337 001204      DEC      REPT      ;DONE YET
2991 013534 001365      BNE      MRRD2     ;NO LOOP
2992
2993          ;READ SYNC"1"
2994
2995 013536 012777 000005 165360      MOV      #5,ARSMR
2996 013544 012777 000015 165352      MOV      #15,ARSMR
2997 013552 104420      MRCK
2998 013554 133615      133615
2999 013556 104000      HLT
3000
3001          ;READ DATA
3002 013560 005037 001226      MRRD3:  CLR      WORK3      ;CLEAR CLOCK COUNT FOR DATA WD
3003 013564 012705 030114      MOV      #INBUF,R5      ;GET STARTING ADDRESS FOR DATA BUFFER
3004 013570 162705 000002      SUB      #2,R5
3005 013574 012737 000045 001206      MOV      #45,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
3006 013602 012737 002200 001204      MOV      #1152.,REPT    ;SETUP COUNTER TO TRANSFER
3007                                     ;64 WORDS-18X64=1152
3008                                     ;1 CLOCK PER 1 BIT OF DATA
3009 013610 104444      IS:     RBIT      ;GET 1 DATA BIT
3010 013612 104440      CLKR1    ;CLOCK MR REG
3011 013614 104000      HLT      ;MR NOT CORRECT
3012
3013 013616 104442      CLKR0    ;CLOCK MR REG
3014 013620 104000      HLT      ;MR REG NOT CORRECT
3015
3016 013622 005337 001204      DEC      REPT      ;DONE WITH COMPLETE TRANSFER
3017 013626 001370      BNE      IS       ;NO
    
```

```

3018 013630 032737 000400 001170 2$: BIT #BIT8,ONCEE ;DID WE ALREADY DO CRC?
3019 013636 001030 BNE 3$ ;YES
3020 013640 052737 000400 001170 BIS #BIT8,ONCEE ;NO SET CRC FLAG
3021 013646 013737 001206 001174 MOV REPT1,SAVEE ;SAVE REPT1
3022 013654 004737 024430 JSR PC,GENCRC ;GENERATE CRC WORD
3023 ;AND LEAVE IN LOC "WORK"
3024 013660 012702 030114 MOV #INBUF,R2
3025 013664 013737 001174 001206 MOV SAVEE,REPT1 ;RESTORE REPT1
3026 013672 062702 000200 ADD #200,R2 ;STORE CRC WORD AT END OF
3027 013676 013712 001216 MOV WORK,R2 ;INBUF TABLE
3028 013702 010205 MOV R2,R5
3029 013704 162705 000002 SUB #2,R5
3030 013710 012737 000022 001204 MOV #18.,REPT ;SETUP TO TRANSFER 1 WD
3031 013716 000734 BR 1$ ;TRANSFER CRC WD
3032 013720 104446 3$: MCLK1 ;CLOCK MR REG
3033 013722 104420 MRCK ;CHECK MR REG
3034 013724 117611 HLT ;TO EQUAL
3035 013726 104000 ;117611
3036 013730 104450 MCLK0 ;CLOCK MR REG
3037 013732 104420 MRCK ;CHECK MR
3038 013734 003601 3601 ;TO EQUAL
3039 013736 104000 HLT ;3601
3040 013740 104446 MCLK1 ;CLOCK MR REG
3041 013742 104420 MRCK ;CHECK MR
3042 013744 113611 HLT ;TO EQUAL
3043 013746 104000 ;113611
3044 013750 104450 MCLK0 ;CLOCK MR REG
3045 013752 104420 MRCK ;CHECK MR
3046 013754 003601 3601 ;TO EQUAL
3047 013756 104000 HLT ;3601
3048
3049 ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
3050 ;STEP INTO END OF SECTOR DEAD BAND
3051 ;EBL IS NOW ASSERTED
3052
3053 013760 012737 000020 001204 MRD4: MOV #20,REPT
3054 013766 104446 1$: MCLK1 ;CLOCK MR REG
3055 013770 104420 MRCK ;CHECK MR REG
3056 013772 113611 HLT ;TO EQUAL
3057 013774 104000 ;113611
3058 013776 104450 MCLK0 ;CLOCK MR REG
3059 014000 104420 MRCK ;CHECK MR
3060 014002 003601 3601 ;REG TO
3061 014004 104000 HLT ;EQUAL 3601
3062 014006 005337 001204 DEC REPT ;DONE YET?
3063 014012 001365 BNE 1$ ;NO
3064
3065 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3066 ;SHOULD GET STROBE BUFFER
3067
3068 014014 104446 MCLK1 ;CLOCK MR REG
3069 014016 104420 MRCK ;CHECK MR
3070 014020 117611 HLT ;REG TO
3071 014022 104000 ;EQUAL 117611

```

K06

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST55

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 76
MAINTENANCE READ TEST

```

3072          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3073          ;SHOULD COMPLETE TRANSFER.
3074
3075 014024 104450 MR05: MCLK0          ;CLOCK MR REG
3076 014026 022777 004270 165044 CMP      #4270, @RSCS1 ;ANY ERRORS?
3077 014034 001401 BEQ      1$          ;NO
3078 014036 104054 HLT      !DA!DS!WC
3079 014040 005777 165040 1$: TST      @RSWC          ;DID WC GO TO 0
3080 014044 001401 BEQ      +4          ;YES
3081 014046 104010 HLT      !WC          ;WC REG SHOULD=0
3082 014050 022777 000001 165032 CMP      #1, @RSDA ;DOES RSD=1
3083 014056 001401 BEQ      +4          ;YES
3084 014060 104004 HLT      !DA          ;NO RSDA SHOULD=1
3085
3086          ;COMPARE DATA READ WITH INPUT BUFFER
3087          ;WILL ONLY TYPEOUT 10 ERRORS --- BUT IF SW12 IS SET
3088          ;IT WILL TYPE OUT ALL ERRORS
3089
3090 014062 042737 000040 001170 MR06: BIC      #BITS, ONCEE ;CLEAR CLOCK TYPEOUT
3091 014070 012701 030114 MOV      #INBUF, GOOD ;GET STARTING LOC OF EXPECTED DATA
3092 014074 012700 030714 MOV      #OUTBUF, BAD ;GET STARTING LOC OF DATA "READ" FROM DISK
3093 014100 012737 000012 001204 MOV      #12, REPT ;SET UP ERROR COUNTER
3094 014106 012705 000101 MOV      #101, R5 ;COMPARE 1 SECTOR
3095 014112 005305 3$: DEC      R5 ;DONE WITH SECTOR
3096 014114 001445 BEQ      2$          ;YES GET OUT
3097 014116 022021 CMP      (BAD)+, (GOOD)+ ;IS DATA CORRECT?
3098 014120 001774 BEQ      3$          ;YES
3099 014122 032777 010000 164676 BIT      #BIT12, @SWR ;TYPE ALL ERRORS?
3100 014130 001003 BNE      1$          ;YES
3101 014132 005337 001204 DEC      REPT ;TYPED OUT 10 ERRORS YET?
3102 014136 001434 BEQ      2$          ;YES GET OUT
3103 014140 024041 1$: CMP      -(BAD), -(GOOD) ;GET ERROR
3104 014142 010003 MOV      BAD, R3 ;SAVE CONTENTS OF
3105 014144 010104 MOV      GOOD, R4 ;GOOD AND BAD
3106 014146 011300 MOV      (R3), BAD ;GET DATA & PLEASE
3107 014150 011401 MOV      (R4), GOOD ;IN GOOD AND BAD
3108 014152 104000 HLT          ;TYPE OUT ERROR
3109 014154 010300 MOV      R3, BAD ;PUT ADDRESS BACK
3110 014156 010401 MOV      R4, GOOD ;INTO GOOD & BAD
3111 014160 010037 001216 MOV      BAD, WORK
3112 014164 032777 020000 164634 BIT      #BIT13, @SWR
3113 014172 001014 BNE      4$          ;
3114 014174 104402 014200 TYPE      +2          ;ASCIZ "BAD ADDRESS= "
3115 014216 013746 001216 MOV      WORK, -(6) ;PUT WORK ON STACK
3116 014222 104406 TYPES          ;TYPE STACK IN OCTAL - SUPPRESS
3117 014224 022021 4$: CMP      (BAD)+, (GOOD)+
3118 014226 000731 BR       3$          ;
3119 014230 2$:          ;DONE

```

```

3120 ;*****
3121 ;TEST 56 MAINTENANCE MODE DATA WRITE CHECK TEST
3122 ;*****
3123 014230 104400 TST56: SCOPE
3124
3125 ;MODULE TESTED: M7771, M7753, M7751
3126 ;A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION.
3127 ;WITHIN THE RS03, A WRITE CHECK FUNCTION IS IDENTICAL TO A
3128 ;READ FUNCTION.
3129
3130 014232 104414 MRWCK: CLRDK ;CLEAR DRIVE REGISTERS
3131 014234 052737 000040 001170 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
3132 C.4242 042737 047716 001170 BIC #47716,ONCEE ;CLEAR ALL OTHER FLAG BITS
3133 014250 104430 MRIND ;SEND INDEX PULSE TO MR REG
3134 014252 104420 MRCK ;CHECK MR REG
3135 014254 022701 22701 ;TO EQUAL 22701
3136 014256 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3137 ;BY SENDING 2 CLOCK PULSES
3138
3139 014260 012737 000004 001144 MOV #4,FLAG2 ;SET WC FLAG FOR CLKR1 ROUTINE
3140
3141 ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
3142 ;DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
3143 ; A WORD OF ALL 1'S
3144 ; FLOATING 1'S PATTERN (16 WORDS)
3145 ; A PATTERN OF 146314 (46 WORDS)
3146
3147 014266 012702 030114 MOV #INBUF,R2 ;GET LOCATION OF INBUF
3148 014272 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
3149 014274 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
3150 014300 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
3151 014302 000261 SEC ;A PATTERN OF FLOATING ONES
3152 014304 006103 1$: ROL R3 ;GET PATTERN
3153 014306 103402 BCS 2$ ;DONE GET OUT
3154 014310 010322 MOV R3,(R2)+ ;FILL BUFFER
3155 014312 000774 BR 1$ ;CONT
3156 014314 012703 000056 2$: MOV #46,R3 ;FILL REMAINING PORTION OF
3157 014320 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
3158 014324 010422 3$: MOV R4,(R2)+ ;LOAD BUFFER
3159 014326 005303 DEC R3 ;DONE YET
3160 014330 001375 BNE 3$ ;NO
3161
3162 ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3163
3164 014332 012777 030114 164546 MOV #INBUF,ARSBA ;LOAD BUS ADDR REG
3165 014340 012777 177700 164536 MOV #177700,ARSWC ;LOAD WORD COUNT REG
3166 014346 012777 000051 164524 MOV #51,ARSCS1 ;LOAD WRITE CHECK COMMAND
3167 014354 104454 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3168 ;TO CLEAR OUT COUNTERS AND REGISTERS
3169 ;THAT OTHERWISE COULD NOT BE CLEARED.
3170 014356 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3171 014360 104456 SPASS ;CLOCK MR REG SP = 1

```

M06

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST56

RS11-R503 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE MODE DATA WRITE CHECK TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 78

```

3172                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3173 014362 104430    MRIND
3174 014364 104420    MRCK                ;CHECK MR REG TO EQUAL
3175 014366 022701    22701                ;22701
3176 014370 104000    HLT
3177
3178                ;STEP THRU RESYNC PERIOD
3179
3180 014372 012737 001000 001204    MOV      #512.,REPT
3181 014400 052737 000040 001170    BIS      #BITS,ONCEE                ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3182 014406 104446    MRWCK1: MCLK1                ;CLOCK MR REG
3183 014410 104420    MRCK                ;CHECK FOR
3184 014412 032711    32711                ;CORRECT DATA
3185 014414 104000    HLT                ;MR=BAD GOOD=CORRECT DATA
3186 014416 104450    MCLK0                ;CLOCK MR REG
3187 014420 104420    MRCK                ;CHECK FOR
3188 014422 022701    22701                ;CORRECT DATA
3189 014424 104000    HLT                ;ERROR WHILE CLOCKING THROUGH RESYNC
3190 014426 005337 001204    DEC      REPT                ;FINISH LOOPING
3191 014432 001365    BNE      MRWCK1                ;THROUGH RESYNC PERIOD
3192
3193                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3194                ;SP=C EQUALS SECTOR PULSE
3195 014434 104446    MCLK1                ;CLOCK MR REG
3196 014436 104420    MRCK                ;MR SHOULD
3197 014440 032311    32311                ;EQUAL 32311
3198 014442 104000    HLT                ;MR=BAD GOOD=CORRECT ANS
3199 014444 104450    MCLK0                ;CLOCK MR REG
3200 014446 104420    MRCK                ;CHECK MR
3201 014450 022301    22301                ;TO EQUAL 22301
3202 014452 104000    HLT                ;MR=BAD GOOD=CORRECT ANS
3203
3204                ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3205
3206 014454 012737 000107 001204    MOV      #71.,REPT
3207 014462 104446    MRWCK2: MCLK1                ;CLOCK MR REG
3208 014464 104420    MRCK                ;CHECK MR REG
3209 014466 033711    33711                ;TO EQUAL 33711
3210 014470 104000    HLT                ;MR=BAD GOOD=CORRECT ANS
3211 014472 104450    MCLK0                ;CLOCK MR REG
3212 014474 104420    MRCK                ;CHECK MR REG
3213 014476 023701    23701                ;TO EQUAL 23701
3214 014500 104000    HLT                ;MR=BAD GOOD=CORRECT ANS
3215 014502 005337 001204    DEC      REPT                ;DONE YET
3216 014506 001365    BNE      MRWCK2                ;NO LOOP

```

```

3217                               ;READ SYNC"1"
3218
3219 014510 012777 000005 164406      MOV      #5,DRSMR
3220 014516 012777 000015 164400      MOV      #15,DRSMR
3221 014524 104420                    MRCK
3222 014526 133715                    133715
3223 014530 104000                    HLT
3224
3225                               ;READ DATA
3226 014532 005037 001226      MRWCK3: CLR      WORK3      ;CLEAR CLOCK COUNT FOR DATA WD
3227 014536 012705 030114      MOV      #INBUF,R5      ;GET STARTING ADDRESS FOR DATA BUFFER
3228 014542 162705 000002      SUB      #2,R5
3229 014546 012737 000045 001206      MOV      #45,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
3230 014554 012737 002200 001204      MOV      #1152.,REPT    ;SETUP COUNTER TO TRANSFER
3231                                     ;64 WORDS-18X64=1152
3232                                     ;1 CLOCK PER 1 BIT OF DATA
3233 014562 104444                    15.  RBIT          ;GET 1 DATA BITS
3234 014564 104440                    CLKR1          ;CLOCK MR REG
3235 014566 104000                    HLT          ;MR NOT CORRECT
3236
3237 014570 104442                    CLKRO
3238 014572 104000                    HLT          ;CLOCK MR REG
3239                                     ;MR REG NOT CORRECT
3240 014574 005337 001204      DEC      REPT          ;DONE WITH COMPLETE TRANSFER
3241 014600 001370                    15.  BNE          ;NO
3242 014602 032737 000400 001170 25: BIT      #BIT8,ONCEE ;DID WE ALREADY DO CRC?
3243 014610 001030                    35.  BNE          ;YES
3244 014612 052737 000400 001170      BIS      #BIT8,ONCEE ;NO SET CRC FLAG
3245 014620 013737 001206 001174      MOV      REPT1,SAVEE   ;SAVE REPT1
3246 014626 004737 024430      JSR      PC,GENCRC    ;GENERATE CRC WORD
3247                                     ;AND LEAVE IN LOC "WORK"
3248 014632 012702 030114      MOV      #INBUF,R2
3249 014636 013737 001174 001206      MOV      SAVEE,REPT1  ;RESTORE REPT1
3250 014644 062702 000200      ADD      #200,R2      ;STORE CRC WORD AT END OF
3251 014650 013712 001216      MOV      WORK,DR2    ;INBUF TABLE
3252 014654 010205                    MOV      R2,R5
3253 014656 162705 000002      SUB      #2,R5
3254 014662 012737 000022 001204      MOV      #18.,REPT    ;SETUP TO TRANSFER 1 WD
3255 014670 000734                    BR       15.          ;TRANSFER CRC WD

```

```

3256 014672 104446          3$:  MCLK1      ;CLOCK MR REG
3257 014674 104420          MRCK      ;CHECK MR REG
3258 014676 117711          117711   ;TO EQUAL
3259 014700 104000          HLT      ;117711
3260 014702 104450          MCLKO    ;CLOCK MR REG
3261 014704 104420          MRCK      ;CHECK MR
3262 014706 003701          3701     ;TO EQUAL
3263 014710 104000          HLT      ;3701
3264 014712 104446          MCLK1    ;CLOCK MR REG
3265 014714 104420          MRCK      ;CHECK MR
3266 014716 113711          113711   ;TO EQUAL
3267 014720 104000          HLT      ;113711
3268 014722 104450          MCLKO    ;CLOCK MR REG
3269 014724 104420          MRCK      ;CHECK MR
3270 014726 003701          3701     ;TO EQUAL
3271 014730 104000          HLT      ;3701

```

```

;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
;STEP INTO END OF SECTOR DEAD BAND
;FBL IS NOW ASSERTED

```

```

3277 014732 012737 000020 001204 MRWCK4: MOV      #20,REPT
3278 014740 104446          1$:  MCLK1    ;CLOCK MR REG
3279 014742 104420          MRCK      ;CHECK MR REG
3280 014744 113711          113711   ;TO EQUAL
3281 014746 104000          HLT      ;113711
3282 014750 104450          MCLKO    ;CLOCK MR REG
3283 014752 104420          MRCK      ;CHECK MR
3284 014754 003701          3701     ;REG TO
3285 014756 104000          HLT      ;EQUAL 3701
3286 014760 005337 001204  DEC      REPT  ;DONE YET?
3287 014764 001365          BNE      1$   ;NO

```

```

;PERFORM ONE MAINTENANCE CLOCK OPERATION
;SHOULD GET STROBE BUFFER

```

```

3292 014766 104446          MCLK1    ;CLOCK MR REG
3293 014770 104420          MRCK      ;CHECK MR
3294 014772 117711          117711   ;REG TO
3295 014774 104000          HLT      ;EQUAL 117711

```

```

;PERFORM ONE MAINTENANCE CLOCK OPERATION
;SHOULD COMPLETE TRANSFER.

```

```

3300 014776 104450          MRWCK5: MCLKO  ;CLOCK MR REG
3301 015000 022777 004250 164072  CMP      #4250, @RSCS1 ;ANY ERRORS?
3302 015006 001401          BEQ      1$   ;NO
3303 015010 104054          HLT      !DA!DS!WC
3304 015012 005777 164066          1$:  TST      @RSWC   ;DID WC GO TO 0
3305 015016 001401          BEQ      +4   ;YES
3306 015020 104010          HLT      !WC   ;WC REG SHOULD=0
3307 015022 022777 000001 164060  CMP      #1, @RSDA  ;DOES RSDA=1
3308 015030 001401          BEQ      +4   ;YES
3309 015032 104004          HLT      !DA   ;RSDA SHOULD=1

```


3310
3311
3312
3313 015034 104400
3314

:TEST 57 MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)

TST57: SCOPE

3315
3316
3317
3318
3319
3320
3321
3322
3323

:MODULES TESTED: M7753;
:THE RS03 DISK IS SET UP TO READ (IN MAINTENANCE MODE) ONE
:SECTOR OF A SPECIALLY CREATED DATA PATTERN WHICH LEAVES ONLY
:ONE BIT SET IN THE CRC REGISTER PRIOR TO CHECKING THE CRC
:WORD. THE CORRESPONDING CRC WORD IS THEN "READ" RESULTING
:IN NO DCK ERROR. THE DATA PATTERN IS THEN MODIFIED (BY
:SHIFTING) AND THE ENTIRE READ SEQUENCE REPEATED UNTIL ALL
:16 BITS IN THE CRC REGISTER HAVE BEEN CHECKED.

3324 015036 012737 000040 001144
3325 015044 104414
3326 015046 052737 000040 001170
3327 015054 042737 047716 001170
3328 015062 104430
3329 015064 104420
3330 015066 022701
3331 015070 104424

MRCRC: MOV #40,FLAG2 ;CLEAR TST FLAG
CLRDK ;CLEAR DRIVE REGISTERS
BIS #BITS,ONCEE ;TYPE CLOCK COUNT IF ERROR OCCURS
BIC #47716,ONCEE ;CLEAR ALL OTHER FLAG BITS
M'IND ;SEND INDEX PULSE TO MR REG
MRCK ;CHECK MR REG
22701 ;TO EQUAL 22701
MRINT ;INIT MAINT MODE (CLEAR MRSP)
;BY SENDING 2 CLOCK PULSES

3333 015072 032737 000020 001144
3334 015100 001023
3335 015102 012737 000001 001174

BIT #BIT4,FLAG2 ;FIRST TIME THROUGH
BNE 3\$;NO
MOV #1,SAVEE ;LOAD 1ST CRC WORD

3336
3337
3338
3339
3340
3341
3342

:FILL MEMORY DATA BUFFER (INBUF) WITH 1 SECTOR
:CREATE BUFFER WITH 72 WORDS OF 16 BITS WHICH EQUALS THE NO. OF BITS IN 64 18 BITS WORDS
:DATA BUFFER CONTAINS 6 WORDS OF ZEROS
: A WORD OF 236
: A WORD OF 140000
: 64 WORDS OF ZEROS

3343
3344
3345

:IN THIS TEST, ALL 18 BITS OF THE RS03 DATA WORD MUST BE
:MANIPULATED. HENCE A TABLE CONTAINING 1152 BITS (64X18) IS
:REQUIRED INSTEAD OF A TABLE CONTAINING 64 WORDS.

3346 015110 012702 030114
3347 015114 012703 000006
3348 015120 012702
3349 015122 012703
3350 015124 001375
3351 015126 012722 000236
3352 015130 012722 140000
3353 015136 012703 000100
3354 015140 005022
3355 015144 005303
3356 015146 001375

MOV #INBUF,R2 ;GET LOCATION OF INBUF
MOV #6,R3 ;SETUP COUNTER
1\$: CLR (R2)+ ;TO CLEAR THE
DEC R3 ;FIRST 6
BNE 1\$;WORDS
MOV #236,(R2)+ ;LOAD A 236
MOV #140000,(R2)+ ;LOAD A 140000
MOV #64,R3 ;SETUP COUNTER
2\$: CLR (R2)+ ;TO CLEAR THE
DEC R3 ;REMAINING WORDS
BNE 2\$;FOR

```

3357          ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3358
3359 015150 012777 030714 163730 35:  MOV      #OUTBUF,ARSBA ;LOAD BUS ADDR REG
3360 015156 012777 177700 163720      MOV      #177700,ARSWC ;LOAD WORD COUNT REG
3361 015164 012777 000071 163706      MOV      #71,ARSCSI ;LOAD READ COMMAND
3362 015172 012702 000200          MOV      #270,R2
3363 015176 012703 030714          MOV      #0L,BUF,R3
3364 015202 052737 000020 001144      BIS      #BI,4,FLAG2 ;NO SET FLAG FOR 1ST TIME THROUGH TEST
3365 015210 005023          45:  CLR      (R3)+
3366 015212 015302          DEC      R2
3367 015214 001375          BNE     45
3368 015216 104454          GETSP
3369          ;CLOCK ROUTINE TO GET SECTOR PULSE
3370          ;TO CLEAR OUT COUNTERS AND REGISTERS
3371 015220 104220          HLT     !MR ;THAT OTHERWISE COULD NOT BE CLEARED.
3372 015222 104456          SPASS ;COULD NOT SET SECTOR PULSE (0)
3373          ;CLOCK MR REG SP = 1
3374          ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3375 015224 104430          MRIND
3376 015226 104420          MRCK
3377 015230 022601          22601 ;CHECK MR REG TO EQUAL
3378 015232 104000          HLT     ;22601 FOR A
3379          ;READ COMD
3380          ;STEP THRU RESYNC PERIOD
3381
3382 015234 012737 001000 001204          MOV      #512,REPT
3383 015242 052737 000040 001170          BIS      #BITS,ONCEE
3384 015250 104446          MRCRC1: MCLK1 ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3385 015252 104420          MRCK ;CLOCK MR REG
3386 015254 032611          32611 ;CHECK FOR
3387 015256 104000          HLT     ;CORRECT DATA
3388 015260 104450          MCLK0 ;MR=BAD GOOD=CORRECT DATA
3389 015262 104420          MRCK ;CLOCK MR REG
3390 015264 022601          22601 ;CHECK FOR
3391 015266 104000          HLT     ;CORRECT DATA
3392 015270 005337 001204          DEC     REPT ;ERROR WHILE CLOCKING THROUGH RESYNC
3393 015274 001365          BNE     MRCRC1 ;FINISH LOOPING
3394          ;THROUGH RESYNC PERIOD
3395          ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3396          ;SP=0 EQUALS SECTOR PULSE
3397 015276 104446          MCLK1 ;CLOCK MR REG
3398 015278 104420          MRCK ;MR SHOULD
3399 015302 032211          32211 ;EQUAL 32211
3400 015304 104000          HLT     ;MR=BAD GOOD=CORRECT ANS
3401 015306 104450          MCLK0 ;CLOCK MR REG
3402 015310 104420          MRCK ;CHECK MR
3403 015312 022201          22201 ;TO EQUAL 22201
3404 015314 104000          HLT     ;MR=BAD GOOD=CORRECT ANS

```

;PERFORM 71 MAINT CLOCK OPERATIONS--

```

3405
3406
3407 015316 012737 000107 001204      MOV      #71.,REPT
3408 015324 104446      MRCRC2: MCLK1      ;CLOCK MR REG
3409 015326 104420      MRCK      ;CHECK MR REG
3410 015330 033611      33611     ;TO EQUAL 33611
3411 015332 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
3412 015334 104450      MCLK0     ;CLOCK MR REG
3413 015336 104420      MRCK      ;CHECK MR REG
3414 015340 023601      23601     ;TO EQUAL 23601
3415 015342 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
3416 015344 005337 001204      DEC      REPT      ;DONE YET
3417 015350 001365      BNE      MRCRC2    ;NO LOOP
3418
3419
3420 015352 012777 000005 163544 ;READ SYNC"1"
3421 015360 012777 000015 163536      MOV      #5,DRSMR
3422 015366 104420      MOV      #15,DRSMR
3423 015370 133615      MRCK
3424 015372 104000      133615
3425      HLT
3426
3427 015374 005037 001226 ;READ DATA
3428 015400 012705 030114 MRCRC3: CLR      WORK3      ;CLEAR CLOCK COUNT FOR DATA WD
3429 015404 162705 000002      MOV      #INBUF,RS      ;GET STARTING ADDRESS FOR DATA BUFFER
3430 015410 012737 000045 001206      SUB      #2,RS
3431 015416 012737 002200 001204      MOV      #45,REPT1
3432      MOV      #1152.,REPT ;SETUP COUNTER FOR 1ST SB BIT
3433      ;SETUP COUNTER TO TRANSFER
3434      ;64 WORDS-18X64=1152
3435      ;1 CLOCK PER 1 BIT OF DATA
3436 015424 104444      15:      RBIT      ;GET 1 DATA BITS
3437 015426 104440      CLKR1     ;CLOCK MR REG
3438 015432 104442      HLT      ;MR NOT CORRECT
3439 015434 104000      CLKR0     ;CLOCK MR REG
3440      HLT      ;MR REG NOT CORRECT
3441 015436 005337 001204      JEC      REPT
3442 015442 001370      BNE      15

```

3443	015444	032737	000400	001170	25:	BIT	#BIT8, ONCEE	: DID WE ALREADY DO CRC?
3444	015452	001020				BNE	35	: YES
3445	015454	052737	000400	001170		BIS	#BIT8, ONCEE	: NO SET CRC FLAG
3446	015462	012702	030114			MOV	#INBUF, R2	: MOVE CRC
3447	015466	062702	000220			ADD	#220, R2	: WORD TO END OF
3448	015472	013712	001174		45:	MOV	SAVEE, @R2	: INBUF TABLE
3449	015476	010205			55:	MOV	R2, RS	: GET CRC WORD
3450	015500	162705	000002			SUB	#2, RS	
3451	015504	012737	000022	001204		MOV	#18., REPT	: SETUP TO TRANSFER 1 WD
3452	015512	000744				BR	15	: TRANSFER CRC WD
3453	015514	104446			35:	MCLK1		: CLOCK MR REG
3454	015516	104420				MRCK		: CHECK MR REG
3455	015520	117611				117611		: TO EQUAL
3456	015522	104000				HLT		: 117611
3457	015524	104450				MCLK0		: CLOCK MR REG
3458	015526	104420				MRCK		: CHECK MR
3459	015530	003601				3601		: TO EQUAL
3460	015532	104000				HLT		: 3601
3461	015534	104446				MCLK1		: CLOCK MR REG
3462	015536	104420				MRCK		: CHECK MR
3463	015540	113611				113611		: TO EQUAL
3464	015542	104000				HLT		: 113611
3465	015544	104450				MCLK0		: CLOCK MR REG
3466	015546	104420				MRCK		: CHECK MR
3467	015550	003601				3601		: TO EQUAL
3468	015552	104000				HLT		: 3601

```

3469                                     ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
3470                                     ;STEP INTO END OF SECTOR DEAD BAND
3471                                     ;E2L IS NOW ASSERTED.
3472
3473 015554 012737 000020 001204 MRCRC4: MOV      #20,REPT
3474
3475 015562 104446                1$:  MCLK1      ;CLOCK MR REG
3476 015564 104420                MRCK      ;CHECK MR REG
3477 015566 113611                113611   ;TO EQUAL
3478 015570 104000                HLT      ;113611
3479 015572 104450                MCLK0    ;CLOCK MR REG
3480 015574 104420                MRCK     ;CHECK MR
3481 015576 003601                3601    ;REG TO
3482 015600 104000                HLT     ;EQUAL 3601
3483 015602 005337 001204        DEC      REPT ;DONE YET?
3484 015606 001365                BNE     1$  ;NO
3485
3486                                     ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3487                                     ;SHOULD GET STROBE BUFFER
3488
3489 015610 104446                MCLK1    ;CLOCK MR REG
3490 015612 104420                MRCK     ;CHECK MR
3491 015614 117611                117611  ;REG TO
3492 015616 104000                HLT     ;EQUAL 117611
3493
3494                                     ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3495                                     ;SHOULD COMPLETE TRANSFER.
3496
3497 015620 104450                MRCRC5: MCLK0 ;CLOCK MR REG
3498 015622 022777 004270 163250  CMP      #4270,@RSC51 ;ANY ERRORS?
3499 015630 001401                BEQ     1$ ;NO
3500 015632 104054                HLT     !DA!DS!WC
3501 015634 005777 163244        1$:  TST     @R5WC ;DID WC GO TO 0
3502 015640 001401                BEQ     +4 ;YES
3503 015642 104010                HLT     !WC ;WC REG SHOULD=0
3504 015644 006137 001174        ROL     SAVEE ;GET NEXT CRC WORD
3505 015650 103404                BCS     2$ ;DONE - BRANCH
3506 015652 004737 027110        JSR     PC,MDATA ;SHIFT DATA PATTERN
3507 015656 000137 015044        JMP     MRCRC ;RESTART TEST WITH NEW DATA PATTERN
3508 015662

```

```

3509 ;*****
3510 ;TEST 60 MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
3511 ;*****
3512 015662 104400 TST60: SCOPE
3513
3514 ;MODULE TESTED M7753
3515 ;THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA
3516 ;PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE
3517 ;CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ".
3518 ;THIS CAUSES A DCK ERROR. THE READ SEQUENCE IS REPEATED 16
3519 ;TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A
3520 ;DCK ERROR.

```

```

3521 015664 012737 000040 001144 MRDCK: MOV #40,FLAG2 ;CLEAR TST FLAG
3522 015672 104414 CLRDK ;CLEAR DRIVE REGISTERS
3523 015674 052737 000040 001170 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
3524 015702 042737 047716 001170 BIC #47716,ONCEE ;CLEAR ALL OTHER FLAG BITS
3525 015710 104430 MRIND ;SEND INDEX PULSE TO MR REG
3526 015712 104420 MRCK ;CHECK MR REG
3527 015714 022701 22701 ;TO EQUAL 22701
3528 015716 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3529 ;BY SENDING 2 CLOCK PULSES
3530 015720 032737 000020 001144 BIT #BIT4,FLAG2 ;FIRST TIME THROUGH
3531 015726 001023 BNE 3$ ;NO
3532 015730 012737 000001 001174 MOV #1,SAVEE ;LOAD 1ST CRC WORD
3533
3534 ;FILL MEMORY DATA BUFFER (INBUF) WITH 64 WORDS (1 SECTOR)
3535 ;CREATE BUFFER WITH 72 WORDS OF 16 BITS WHICH = THE NO. OF BITS IN 64 18 BIT WORDS
3536 ;DATA BUFFER CONTAINS 7 WORDS OF ZEROS
3537 ; A WORD OF 23
3538 ; A WORD OF 154000
3539 ; 63 WORDS OF ZEROS
3540
3541 015736 012702 030114 MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
3542 015742 012703 000007 MOV #7,R3 ;SETUP COUNTER
3543 015746 005022 1$: CLR (R2)+ ;TO CLEAR THE
3544 015750 005303 DEC R3 ;FIRST 7
3545 015752 001375 BNE 1$ ;WORDS
3546 015754 012722 000023 MOV #23,(R2)+ ;LOAD A 23
3547 015760 012722 154000 MOV #154000,(R2)+ ;LOAD A 154000
3548 015764 012703 000077 MOV #63,R3 ;SETUP COUNTER
3549 015770 005022 2$: CLR (R2)+ ;TO CLEAR THE
3550 015772 005303 DEC R3 ;REMAINING WORDS
3551 015774 001375 BNE 2$ ;FOR THAT SECTOR
3552 ;SETUP CONTROLLER TO TRANSFER 64 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3553
3554 015776 012777 030714 163102 3$: MOV #OUTBUF,ARSBA ;LOAD BUS ADDR REG
3555 016004 012777 177700 163072 MOV #177700,ARSWC ;LOAD WORD COUNT REG
3556 016012 012777 000071 163060 MOV #71,ARSCS1 ;LOAD READ COMMAND
3557 016020 012702 000200 MOV #200,R2
3558 016024 012703 030714 MOV #OUTBUF,R3
3559 016030 052737 000020 001144 BIS #BIT4,FLAG2 ;NO SET FLAG FOR 1ST TIME THROUGH TEST
3560 016036 005023 4$: CLR (R3)+
3561 016040 005302 DEC R2
3562 016042 001375 BNE 4$

```

```

3563 016044 104454          GETSP          ;CLOCK ROUTINE TO GET SECTOR PULSE
3564                                ;TO CLEAR OUT COUNTERS AND REGISTERS
3565                                ;THAT OTHERWISE COULD NOT BE CLEARED.
3566 016046 104220          HLT          !MR          ;COULD NOT SET SECTOR PULSE (0)
3567 016050 104456          SPASS         ;CLOCK MR REG SP = 1
3568                                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3569 016052 104430          MRIND         ;CHECK MR REG TO EQUAL
3570 016054 104420          MRCK          ;22601
3571 016056 022601          22601
3572 016060 104000          HLT
3573
3574                                ;STEP THRU RESYNC PERIOD
3575
3576 016062 012737 001000 001204          MOV          #512.,REPT
3577 016070 052737 000040 001170          BIS          #BITS,ONCEE
3578 016076 104446          MRDCK1: MCLK1
3579 016100 104420          MRCK
3580 016102 032611          32611
3581 016104 104000          HLT
3582 016106 104450          MCLK0
3583 016110 104420          MRCK
3584 016112 022601          22601
3585 016114 104000          HLT
3586 016116 005337 001204          DEC          REPT
3587 016122 001365          BNE          MRDCK1
3588
3589                                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3590                                ;SP=0 EQUALS SECTOR PULSE
3591 016124 104446          MCLK1
3592 016126 104420          MRCK
3593 016130 032211          32211
3594 016132 104000          HLT
3595 016134 104450          MCLK0
3596 016136 104420          MRCK
3597 016140 022201          22201
3598 016142 104000          HLT

```

```

;CLOCK ROUTINE TO GET SECTOR PULSE
;TO CLEAR OUT COUNTERS AND REGISTERS
;THAT OTHERWISE COULD NOT BE CLEARED.
;COULD NOT SET SECTOR PULSE (0)
;CLOCK MR REG SP = 1
;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
;CHECK MR REG TO EQUAL
;22601
;STEP THRU RESYNC PERIOD
;TYPE OUT CLOCK COUNT
;CLOCK MR REG
;CHECK FOR
;CORRECT DATA
;MR=BAD GOOD=CORRECT DATA
;CLOCK MR REG
;CHECK FOR
;CORRECT DATA
;ERROR WHILE CLOCKING THROUGH RESYNC
;FINISH LOOPING
;THROUGH RESYNC PERIOD
;CLOCK MR REG
;MR SHOULD
;EQUAL 32211
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR
;TO EQUAL 22201
;MR=BAD GOOD=CORRECT ANS

```

```

3599          ;PERFORM 71 MAINT CLOCK OPERATIONS--
3600
3601 016144 012737 000107 001204      MOV      #71.,REPT
3602 016152 104446      MRDCK2: MCLK1      ;CLOCK MR REG
3603 016154 104420      MRCK        ;CHECK MR REG
3604 016156 033611      33611      ;TO EQUAL 33611
3605 016160 104000      HLT        ;MR=BAD GOOD=CORRECT ANS
3606 016162 104450      MCLKO      ;CLOCK MR REG
3607 016164 104420      MRCK        ;CHECK MR REG
3608 016166 023601      23601      ;TO EQUAL 23601
3609 016170 104000      HLT        ;MR=BAD GOOD=CORRECT ANS
3610 016172 005337 001204      DEC      REPT
3611 016176 001365      BNE      MRDCK2  ;DONE YET
3612
3613
3614 016200 012777 000005 162716      ;READ SYNC"1"
3615 016206 012777 000015 162710      MOV      #5,RSMR
3616 016214 104420      MOV      #15,RSMR
3617 016216 133615      MRCK
3618 016220 104000      133615
3619      HLT
3620
3621 016222 005037 001226      ;READ DATA
3622 016226 012705 030114      MRDCK3: CLR      WORK3      ;CLEAR CLOCK COUNT FOR DATA WD
3623 016232 162705 000002      MOV      #INBUF,R5      ;GET STARTING ADDRESS FOR DATA BUFFER
3624 016236 012737 000045 001206      SUB      #2,R5
3625 016244 012737 002200 001204      MOV      #45,REPT1
3626      MOV      #1152.,REPT      ;SETUP COUNTER FOR 1ST SB BIT
3627      ;SETUP COUNTER TO TRANSFER
3628 016252 104444      15:      RBIT        ;64 WORDS-18X64=1152
3629 016254 104440      CLKR1      ;1 CLOCK PER 1 BIT OF DATA
3630 016256 104000      HLT        ;GET 1 DATA BITS
3631      ;CLOCK MR REG
3632 016260 104442      CLKRO      ;MR NOT CORRECT
3633 016262 104000      HLT        ;CLOCK MR REG
3634      ;MR REG NOT CORRECT
3635 016264 005337 001204      DEC      REPT
3636 016270 001370      BNE      15      ;DONE WITH COMPLETE TRANSFER
3637      ;NO

```


K07

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST60

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 89
MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

3637	016272	032737	000400	001170	2\$:	BIT	#BIT8,ONCEE	:DID WE ALREADY DO CRC?
3638	016300	001020				BNE	3\$:YES
3639	016302	052737	000400	001170		BIS	#BIT8,ONCEE	:NO SET CRC FLAG
3640	016310	012702	030114			MOV	#INBUF,R2	:MOVE CRC
3641	016314	062702	000220			ADD	#220,R2	:WORD TO END OF
3642	016320	012712	000000		4\$:	MOV	#0,R2	:INBUF TABLE
3643	016324	010205			5\$:	MOV	R2,R5	:GET CRC WORD
3644	016326	162705	000002			SUB	#2,R5	
3645	016332	012737	000022	001204		MOV	#18.,REPT	:SETUP TO TRANSFER 1 WD
3646	016340	000744				BR	1\$:TRANSFER CRC WD
3647	016342	104446			3\$:	MCLK1		:CLOCK MR REG
3648	016344	104420				MRCK		:CHECK MR REG
3649	016346	117611				117611		:TO EQUAL
3650	016350	104000				HLT		:117611
3651	016352	104450				MCLK0		:CLOCK MR REG
3652	016354	104420				MRCK		:CHECK MR
3653	016356	003601				3601		:TO EQUAL
3654	016360	104000				HLT		:3601
3655	016362	104446				MCLK1		:CLOCK MR REG
3656	016364	104420				MRCK		:CHECK MR
3657	016366	113611				113611		:TO EQUAL
3658	016370	104000				HLT		:113611
3659	016372	104450				MCLK0		:CLOCK MR REG
3660	016374	104420				MRCK		:CHECK MR
3661	016376	003601				3601		:TO EQUAL
3662	016400	104000				HLT		:3601

```

3663 ;PERFORM 20 MAINTENANCE CLOCK OPERATIONS
3664 ;STEP INTO END OF SECTOR DEAD BAND
3665 ;EBL IS NOW ASSERTED
3666
3667 016402 012737 000020 001204 MRDCK4: MOV #20,REPT
3668
3669 016410 104446 1S: MCLK1 ;CLOCK MR REG
3670 016412 104420 MRCK ;CHECK MR REG
3671 016414 113611 113611 ;TO EQUAL
3672 016416 104000 HLT ;113611
3673 016420 104450 MCLK0 ;CLOCK MR REG
3674 016422 104420 MRCK ;CHECK MR
3675 016424 003601 3601 ;REG TO
3676 016426 104000 HLT ;EQUAL 3601
3677 016430 005337 001204 DEC REPT ;DONE YET?
3678 016434 001365 BNE 1S ;NO
3679
3680 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3681 ;SHOULD GET STROBE BUFFER
3682
3683 016436 104446 MCLK1 ;CLOCK MR REG
3684 016440 104420 MRCK ;CHECK MR
3685 016442 117611 117611 ;REG TO
3686 016444 104000 HLT ;EQUAL 117611
3687
3688 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3689 ;SHOULD COMPLETE TRANSFER.
3690
3691 016446 104450 MRDCK5: MCLK0 ;CLOCK MR REG
3692 016450 022777 144270 162422 CMP #144270, @RSCS1 ;ANY ERRORS?
3693 016456 001401 BEQ 1S ;NO
3694 016460 104054 HLT !DA!DS!WC
3695 016462 005777 162416 1S: TST @RSWC ;DID WC GO TO 0
3696 016466 001401 BEQ .+4 ;YES
3697 016470 104010 HLT :WC ;WC REG SHOULD=0
3698 016472 022777 100000 162414 CMP #100000, @RSER ;DID DCK SET?
3699 016500 001417 BEQ 3S ;YES
3700 016502 104050 HLT !DS!WC
3701 016504 104402 016510 TYPE .+2 ;.ASCIZ <15><12>"DCK DID NOT SET "
3702 016534 004737 023006 JSR PC,CRCTYP ;GET IC THAT FAILED AND TYPE IT
3703 016540 000241 3S: CLC
3704 016542 006137 001174 ROL SAVEE ;GET NEXT CRC WORD
3705 016546 103404 BCS 2S ;DONE - BRANCH
3706 016550 004737 027110 JSR PC,MDATA ;SHIFT DATA PATTERN
3707 016554 000137 015672 JMP MRDCK ;RESTART TEST WITH NEW DATA PATTERN
3708 016560 2S: ;DONE

```

```

3709 ;*****
3710 ;TEST 61          IGNORE FUNCTION TEST
3711 ;*****
3712 016560 10440C TST61: SCOPE
3713
3714 ;MODULE TESTED: M7759, M7770
3715 ;PUT THE DISK IN MAINTENANCE MODE AND SET ERROR CONDITIONS IN THE DRIVE
3716 ;ERROR REGISTER (RSER). TRY TO START A READ TRANSFER. THE "GO" BIT IN
3717 ;RSCS1 SHOULD NOT SET. MISSED TRANSFER ERROR (MXF) SHOULD SET IN RSCS2
3718 ;WHICH IN TURN SHOULD CAUSE "TRE" AND "SC" TO SET IN RSCS1.
3719
3720 016562 104414 MRIFT: CLRDK ;CLEAR ALL REGISTERS
3721 016564 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
3722 016566 104420 MRCK ;CHECK MR REG
3723 016570 022701 22701 ;CHECK MR REG
3724 016572 104424 MRINT ;TO EQUAL 22701
3725 016574 012777 177777 162312 MOV #-1,RSER ;INIT MAINT MODE (CLEAR MRSP)
3726 016602 013777 001164 162306 MOV UNITSV,RSAS ;SET ERRORS
3727 ;CLEAR ATA BIT IN RSAS
3728 016610 012777 030714 162270 MOV #OUTBUF,RSBA ;AND ERROR BITS IN RSCS1
3729 016616 012777 177777 162260 MOV #-1,RSWC ;LOAD RSBA
3730 016624 012777 000071 162246 MOV #71,RSRCS1 ;LOAD RSWC
3731 016632 032777 000001 162240 BIT #BIT0,RSRCS1 ;LOAD READ FUNCTION
3732 016640 001401 BEQ IS ;IS "GO" BIT ZERO?
3733 016642 104140 HLT !DS!AS ;YES
3734 ;"GO" BIT IN RSCS1 SHOULD NOT
3735 016644 012737 177777 001216 1S: MOV #177777,WORK ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
3736 016652 005337 001216 5S: DEC WORK ;SETUP TIMEOUT FOR MXF ERROR
3737 016656 000240 NOP
3738 016660 000240 NOP
3739 016662 001373 BNE 5S
3740 016664 017700 162212 MOV RSRCS2,BAD ;CHECK RSCS2 FOR MXF
3741 016670 012701 001100 MOV #1100,GOOD ;GET CORRECT ANS
3742 016674 053701 001162 BIS UNNUM,GOOD ;FOR RSCS2
3743 016700 020001 CMP BAD,GOOD ;IS RSCS2 CORRECT
3744 016702 001401 BEQ 2S ;YES
3745 016704 104000 HLT ;BAD=RSCS2 GOOD=CORRECT ANS
3746 ;MXF SHOULD BE SET IN RSCS2
3747 ;FOR A READ WAS ISSUED
3748 ;WITH ERROR BITS SET IN RSER.
3749 016706 022777 144270 162164 2S: CMP #144270,RSRCS1 ;IS RSCS1 CORRECT?
3750 016714 001401 BEQ 3S ;YES
3751 016716 104042 HLT !DS!ER ;SC AND TRE SHOULD BE SET FOR
3752 ;MXF SHOULD BE SET IN RSCS2

```



```

3777 :*****
3778 :TEST 62          INVALID ADDRESS ERROR (IAE) TEST
3779 :*****
3790 017000 104400 TST62: SCOPE
3781
3782 :MODULE TESTED M7754, M7770
3783 :FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
3784 :ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN
3785 :THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
3786 :RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
3787 :DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
3788 :CONTROL REGISTER (RSCS1).
3789 017002 042737 000040 001170      BIC      #BIT5,ONCEE      ;CLEAR CLK CNT FLAG
3790 017010 012702 004000              MOV      #4000,R2      ;LOAD R2 WITH INVALID ADDR
3791 017014 012737 017022 001010      MOV      #45,LAD       ;LOOP HERE ON ERROR
3792 017022 104416      4S:  MROMD          ;PUT DRIVE IN MAINT MODE
3793 017024 104420      MRCK           ;CHECK MAINT REG
3794 017026 022701      MRINT          ;INIT MAINT MODE (CLEAR MRSP)
3795 017030 104424      MRINT          ;LOOPING ON ERRORS)
3796 017032 032737 000004 001170      BIT      #BIT2,ONCEE  ;YES
3797 017040 001002      BNE              ;GET INVALID ADDRESS
3798 017042 006102      ROL      R2        ;DONE FLOATING A ONE YET?
3799 017044 103454      BCS      IADONE    ;LOAD RSDA WITH INVALID ADDRESS
3800 017046 010277 162036 162020 1S:  MOV      R2,RSDA    ;DO A READ TO INVALID ADDR
3801 017052 012777 000071 162020      MOV      #71,RSCS1  ;IS RSER CORRECT?
3802 017060 022777 002000 162026      CMP      #2000,RSER ;YES
3803 017066 001404      BEQ              ;SET ERROR BIT
3804 017070 052737 000004 001170      BIS      #BIT2,ONCEE ;RSER SHOULD=2000 FOR
3805 017076 104044      HLT      !DS!DA    ;A READ COMMAND WAS GIVEN
3806 017100 042737 000004 001170 2S:  BIC      #BIT2,ONCEE  ;TO AN ILLEGAL ADDRESS
3807 017106 022777 150600 161776      CMP      #150600,RSDS ;CLEAR ERROR FLAG
3808 017114 001404      BEQ              ;DID IAE SET?
3809 017116 052737 000004 001170      BIS      #BIT2,ONCEE ;YES
3810 017124 104044      HLT      !DS!DA    ;SET ERROR BIT
3811 017126 042737 000004 001170 3S:  BIC      #BIT2,ONCEE  ;RSDS SHOULD=150600 FOR
3812 017134 022777 144270 161736      CMP      #144270,RSRCS1 ;IAE SHOULD BE SET IN RSER
3813 017142 001404      BEQ              ;CLEAR ERROR FLAG
3814 017144 052737 000004 001170      BIS      #BIT2,ONCEE  ;DID SC + TRE SET?
3815 017152 104044      HLT      !DA!DS    ;YES
3816 017154 042737 000004 001170 4S:  BIC      #BIT2,ONCEE  ;SET ERROR BIT
3817 017162 104414      CLDRK          ;SC + TRE SHOULD BE SET IN RSCS1
3818 017164 005777 161724      TST      @RSER      ;FOR IAE SHOULD BE SET IN RSER
3819 017170 001401      BEQ      +4        ;CLEAR ERROR BIT
3820 017172 104040      HLT      !DS      ;CLEAR ALL ERRORS
3821 017174 000712      BR         4S      ;DID IAE CLEAR?
3822 :*****
3823 :
3824 :
3825 :
3826 :
IADONE: ;CONTINUE
;DONE

```

```

3827 ;*****
3828 ;TEST 63 OPERATION INCOMPLETE ERROR TEST
3829 ;*****
3830 017176 104400 TST63: SCOPE
3831
3832 ;MODULE TESTED M7770
3833 ;PUT THE DISK IN MAINTENANCE MODE AND START A READ COMMAND
3834 ;THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE
3835 ;ROTATION OF THE DISK SURFACE. THE THIRD INDEX PULSE SHOULD
3836 ;CAUSE OPERATION IN COMPLETE "OPI" TO APPEAR IN THE DRIVE ERROR
3837 ;REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSDS)
3838
3839 017200 104414 MROPI: CLROK ;CLEAR ALL DRIVE REGISTERS
3840 017202 013777 030714 161676 MOV 2,OUTBUF,RSBA ;SETUP RSBA
3841 017210 012777 177777 161666 MOV #-1,RSWC ;SETUP RSWC
3842
3843 017216 104416 MROMD ;PUT DRIVE INTO MAINT MODE
3844 017220 104420 MRCK ;CHECK MAINT REG
3845 017222 022701 22701 ;TO EQUAL 22701
3846 017224 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3847
3848 017226 012777 00071 161644 MOV #71,RSOS1 ;LOAD A READ COMMAND
3849
3850 017234 104430 MRIND ;ISSUE THREE INDEX
3851 017236 104430 MRIND ;PULSES TO
3852 017240 104430 MRIND ;CAUSE OPI
3853
3854 ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
3855 017242 017700 161646 MOV 2,RSER,BAD ;GET RSER
3856 017246 012701 020000 MOV #2000,GOOD ;GET CORRECT ANS
3857 017252 020100 CMP GOOD,BAD ;DID OPI SET IN RSER?
3858 017254 001434 BEQ IS ;YES
3859 017256 104402 017262 TYPE ,,+2 ;ASCIZ (15)(12)"OPI IN RSER SHOULD SET-3 INDEX PULSES W
3860 017344 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
3861
3862 017346 022777 150600 161536 IS: CMP #150600,RSOS ;DID CORRECT ERRORS SET?
3863 017354 001401 BEQ 2S ;YES
3864 017356 104040 HLT !DS ;RSDS SHOULD=150600 BECAUSE
3865 ;OF OPI ERROR IN RSER
3866 017360 022777 144270 161512 2S: CMP #144270,RSOS1 ;DID SC AND TRE SET IN RSCS1?
3867 017366 001401 BEQ MROPIA ;YES
3868 017370 104050 HLT !DS!WC ;SC AND TRE SHOULD SET IN RSCS1
3869 ;BECAUSE OF ERROR IN RSER
3870 017372 104414 MROPIA: CLROK ;CLEAR ALL ERRORS
3871 017374 005777 161514 TST 2,RSER ;DID OPI CLEAR IN RSER ?
3872 017400 001437 BEQ IS ;YES
3873 017402 104402 017406 TYPE ,,+2 ;ASCIZ (15)(12)"OPI IN RSER DID NOT CLEAR BY SETTING CL
3874 017476 104040 HLT !DS ;RSER SHOULD=0
3875 017500 022777 010600 161404 IS: CMP #10600,RSOS ;DID ERROR BITS CLEAR IN RSDS
3876 ;BY SETTING CLR BIT IN RSCS2
3877 017506 001401 BEQ ,+4 ;YES
3878 017510 104040 HLT !DS ;RSDS SHOULD=10600

```

```

3879 :*****
3880 :TEST 64 PARITY ERROR TEST
3881 :*****
3882 017512 104400 TST64: SCOPE
3883
3884 :MODULES TESTED: M7754, M7770
3885 :SET "PAT" BIT IN RSCS2. WRITE A DRIVE REGISTER. "PAR" SHOULD SET IN
3886 :THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS
3887 :AND 'SC' TO SET IN RSCS1.
3888
3889 017514 104414 MRPAR: CLROK ;CLEAR ALL REGISTERS
3890 017516 042737 000040 0C1170 BIC #BITS,ONCE ;CLEAR CLK CNT FLAG
3891 017524 104416 MROMD ;PUT DRIVE IN MAINT MODE
3892 017526 104420 MRCK ;CHECK MAINT TO
3893 017530 022701 22701 ;EQUAL 22701
3894 017532 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3895 017534 052777 000020 161340 BIS #BIT4,RSCS2 ;SET THE "PAT" BIT.
3896 017542 012777 000077 161340 MOV #77,RSDA ;BY WRITING INTO THIS REGISTER,
3897 ;PAR SHOULD SET IN RSER
3898 017550 022777 000010 161336 CMP #10,RSER ;DID PAR SET?
3899 017556 001401 BEQ +4 ;YES
3900 017560 104044 HLT !DS ;"PAR" IN RSER SHOULD BE SET FOR
3901 ;THE "PAT" BIT WAS SET IN RSCS2
3902 ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
3903 017562 022777 104200 161310 CMP #104200,RSCS1 ;DID PAR CAUSE SC TO SET?
3904 017570 001401 BEQ +4 ;YES
3905 017572 104044 HLT !DS!DA ;SC SHOULD BE SET IN RSCS1 FOR
3906 ;PAR SHOULD BE SET IN RSER
3907 017574 022777 000077 161306 CMP #77,RSDA ;DID RSDA GET LOADED?
3908 017602 001401 BEQ +4 ;YES
3909 017604 104004 HLT !DA ;RSDA SHOULD=77 FOR PAT
3910 ;BIT WAS SET WHEN PROGRAM
3911 ;TRIED TO WRITE INTO RSDA
3912 017606 104414 CLROK ;CLEAR ALL ERRORS
3913 017610 022777 004200 161262 CMP #4200,RSCS1 ;DID ERRORS CLEAR?
3914 017616 001401 BEQ +4 ;YES
3915 017620 104044 HLT !DS!DA ;SC DID NOT CLEAR BY USING
3916 ;THE "CLR" BIT IN RSCS2
3917 017622 005777 161266 TST RSER ;DID PAR CLEAR?
3918 017626 001401 BEQ +4 ;YES
3919 017630 104044 HLT !DS!DA ;PAR DID NOT CLEAR BY USING
3920 ;THE CLR BIT IN RSCS2

```

```

3921 :*****
3922 :TEST 65 MAINTENANCE MODE INTERRUPT TEST
3923 :*****
3924 017632 104400 †ST65: SCOPE
3925
3926 :MODULE TESTED M7771
3927 :IN THIS TEST THE INTERRUPT ENABLE BIT IS SET (I.E.).
3928 :A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMR"
3929 :ERROR IS CREATED WHILE THE FIRST SECTOR IS BEING WRITTEN
3930 :THIS SHOULD CAUSE THE DRIVE TO INTERRUPT AFTER THE FIRST
3931 :SECTOR IS WRITTEN. AND CAUSE THE TRANSFER TO TERMINATE
3932
3933 017634 012737 001602 001144 MREX: MOV #16C2,FLAG2
3934 017642 104414 CLRDK ;CLEAR DRIVE REGISTERS
3935 017644 012737 000200 177776 MOV #200,2#PS ;SETUP FOR INTERRUPT
3936 017652 012706 000500 MOV #500,SP
3937 017656 012737 000040 001170 MOV #4C,ONCEE ;SET TYPE CLOCK CNT WITH ERROR MESSAGE FLAG
3938 017664 104430 MRIND ;SEND INDEX PULSE TO MR REG
3939 017666 104420 MRCK ;CHECK MR REG
3940 017670 022701 22701 ;TO EQUAL 22701
3941 017672 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3942 ;BY SENDING 2 CLOCK PULSES
3943
3944 :FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (2 SECTORS)
3945 :DATA BUFFER WORDS ARE : WORD OF ALL 0'S - ALL 1'S
3946 : FLOATING 1'S PATTERN (16 WORDS)
3947 : A PATTERN OF 146314 (110 WORDS)
3948
3949 017674 012702 030114 MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
3950 017700 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
3951 017702 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
3952 017706 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
3953 017710 000261 SEC ;A PATTERN OF FLOATING ONES
3954 017712 006103 15: ROL R3 ;GET PATTERN
3955 017714 103402 BCS 2$ ;DONE GET OUT
3956 017716 010322 MOV R3,(R2)+ ;FILL BUFFER
3957 017720 000774 BR 1$ ;CONT
3958 017722 012703 000156 25: MOV #110,R3 ;FILL REMAINING PORTION OF
3959 017726 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
3960 017732 010422 35: MOV R4,(R2)+ ;LOAD BUFFER
3961 017734 005303 DEC R3 ;DONE YET?
3962 017736 001375 BNE 3$ ;NO
3963
3964 ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (2 SECTORS)
3965 017740 012777 020560 161162 MOV #INTMR,2#SVEC ;SETUP INTERRUPT VECTOR
3966 017746 012777 000340 161156 MOV #340,2#SVCPS
3967 017754 012777 030114 161124 MOV #INBUF,2#SBA ;LOAD BUS ADDR REG
3968 017762 012777 177600 161114 MOV #177600,2#SWC ;LOAD WORD COUNT REG
3969 017770 012777 000161 161102 MOV #161,2#SCSI ;LOAD WRITE COMMAND I/E
3970 017776 104454 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3971 ;TO CLEAR OUT COUNTERS AND REGISTERS
3972 ;THAT OTHERWISE COULD NOT BE CLEARED.
3973 020000 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3974 020002 104456 SPASS ;CLOCK MR REG SP = 1

```



```

3975                                     :ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3976 020004 104430                       MRIND
3977 020006 104420                       MRCK                                     :CHECK MR REG TO EQUAL
3978 020010 020501                       20501                                 :20501
3979 020012 104000                       HLT
3980
3981                                     ;STEP THRU RESYNC PERIOD
3982
3983 020014 012737 001000 001204          MOV      #512.,REPT
3984 020022 052737 000040 001170          BIS      #BITS,ONCEE
3985 020030 104446                       MREX1: MCLK1                               :TYPE OUT CLOCK COUNT IF ERROR OCCURS
3986 020032 104420                       MRCK                               :CLOCK MR REG
3987 020034 030511                       30511                              :CHECK FOR
3988 020036 104000                       HLT                                :CORRECT DATA
3989 020040 104450                       MCLK0                               :MR = BAD GOOD = CORRECT DATA
3990 020042 104420                       MRCK                               :CLOCK MR REG
3991 020044 020501                       20501                              :CHECK FOR
3992 020046 104000                       HLT                                :CORRECT DATA
3993 020050 005337 001204          DEC      REPT                       :ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
3994 020054 001365                       BNE      MREX1                      :FINISH LOOPING
3995                                     :THROUGH RESYNC PERIOD
3996
3997                                     ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3998                                     ;SP=0 EQUALS SECTOR PULSE
3998 020056 104446                       MCLK1                               :CLOCK MR REG
3999 020060 104420                       MRCK                               :MR SHOULD
4000 020062 030111                       30111                              :EQUAL 30111
4001 020064 104000                       HLT                                :MR=BAD GOOD=CORRECT ANS
4002 020066 104450                       MCLK0                               :CLOCK MR REG
4003 020070 104420                       MRCK                               :CHECK MR
4004 020072 020101                       20101                              :TO EQUAL 20101
4005 020074 104000                       HLT                                :MR=BAD GOOD=CORRECT ANS
4006
4007                                     ;PERFORM 63 MAINT CLOCK OPERATIONS--WRITING PREAMBLE
4008
4009 020076 012737 000077 001204          MOV      #63.,REPT
4010 020104 104446                       MREX2: MCLK1                               :CLOCK MR REG
4011 020106 104420                       MRCK                               :CHECK MR REG
4012 020110 031511                       31511                              :TO EQUAL 31511
4013 020112 104000                       HLT                                :MR=BAD GOOD=CORRECT ANS
4014 020114 104450                       MCLK0                               :CLOCK MR REG
4015 020116 104420                       MRCK                               :CHECK MR REG
4016 020120 021501                       21501                              :TO EQUAL 21501
4017 020122 104000                       HLT                                :MR=BAD GOOD=CORRECT ANS
4018 020124 005337 001204          DEC      REPT                       :DONE YET
4019 020130 001365                       BNE      MREX2                      :NO LOOP

```

```

4020                :DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
4021
4022 020132 104446      MCLK1                :CLOCK MR REG
4023 020134 104420      MRCK                 :CHECK MR REG
4024 020136 131511      131511              :TO EQUAL 131511
4025 020140 104000      HLT                 :MR REG=BAD GOOD=CORRECT ANS
4026 020142 104450      MCLK0                :CLOCK MR REG
4027 020144 104420      MRCK                 :MR REG SHOULD
4028 020146 025501      25501              :EQUAL 25501
4029 020150 104000      HLT                 :MR REG=BAD GOOD=CORRECT ANS
4030 020152 104446      MCLK1
4031 020154 104420      MRCK
4032 020156 135511      135511
4033 020160 104000      HLT
4034
4035 020162 012737 000010 001204 ;PERFORM NEXT STEP 8 TIMES TO FINISH WRITING PREAMBLE
4036 020170 104450      MOV                #10,REPT
4037 020172 104420      MREX3: MCLK0                :CLOCK MR REG
4038 020174 025501      MRCK                 :CHECK MR REG
4039 020176 104000      25501              :TO EQUAL 25501
4040 020200 104446      HLT                 :MR=BAD GOOD=CORRECT ANS
4041 020202 104420      MCLK1                :CLOCK MR REG
4042 020204 135511      MRCK                 :CHECK MR REG
4043 020206 104000      135511              :TO EQUAL 135511
4044 020210 005337 001204      HLT                 :MR REG=BAD GOOD=CORRECT ANS
4045 020214 001365      DEC                REPT                :DONE YES?
4046                                     BNE                MREX3                :NO LOOP BACK
4047
4048                ;MOVE DATA WORD INTO RS03 SHIFT REGISTER
4049
4049 020216 104450      MCLK0                :CLOCK MR REG
4050 020220 104420      MRCK                 :CHECK MR REG
4051 020222 021501      21501              :TO EQUAL 21501
4052 020224 104000      HLT                 :MR=BAD GOOD=CORRECT ANS
4053 020226 104446      MCLK1                :CLOCK MR REG
4054 020230 104420      MRCK                 :MR REG SHOULD
4055 020232 123511      123511              :EQUAL 123511
4056 020234 104000      HLT                 :MR=BAD GOOD=CORRECT ANS
4057
4058                ;ENCODE SYNC 1
4059
4060 020236 104450      MCLK0                :CLOCK MR REG
4061 020240 104420      MRCK                 :MR REG SHOULD NOW
4062 020242 033501      33501              :EQUAL 33501
4063 020244 104000      HLT                 :MR=BAD GOOD=CORRECT ANS
4064 020246 012705 030114      MOV                #INBUF,R5                :GET STARTING ADDR FOR DATA BUFFER
4065 020252 011504      MOV                (R5),R4                :GET DATA

```

H08

MAINDEC-11-DZRSE-C
DZRSEC.P11

TST65

RS11-RSG3 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE MODE INTERRUPT TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 99

```

4066 020254 012737 002156 001216      MOV      #1134.,WORK      ;DOING A 1 SECTOR TRANSFER 127 WORDS
4067                                     ;18 BITS PER WORD-CLOCK LOOPS
4068                                     ;TAKE CARE OF 2 BITS AT A TIME
4069                                     ;64 TIMES 18 EQUALS 1134 LOOPS
4070                                     ;TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
4071 020262 052737 000100 001170      BIS      #BIT6,ONCEE     ;SET 1ST TRANSFER WORD FLAG
4072 020270 104432          1$:      XBIT          ;GET 1 BIT OF DATA
4073 020272 104434                   CLKD1       ;SET MCLK
4074                                     ;AND CALCULATE MR REG
4075                                     ;FOR CORRECT DATA (MWOB)
4076 020274 104000          HLT          ;MR REG NOT CORRECT
4077 020276 104436                   CLKD0       ;CLEAR MCLK TO
4078                                     ;COMPLETE TRANSFER OF 1 BIT
4079                                     ;CALCULATE CORRECT ANS FOR
4080                                     ;MR REG (MWOB)
4081 020300 104000          HLT          ;MR=BAD GOOD=CORRECT ANS
4082 020302 032737 000200 001170      BIT      #BIT7,ONCEE     ;ON LAST WORD YET?
4083 020310 001015          BNE      2$           ;YES
4084 020312 032737 000400 001170      BIT      #BIT8,ONCEE     ;ON CRC WORD YET?
4085 020320 001043          BNE      3$           ;YES
4086 020322 005337 001216          DEC      WORK          ;DONE WITH 63 WORDS?
4087 020326 001360          BNE      1$           ;NO
4088
4089 020330 052737 000200 001170      BIS      #BIT7,ONCEE     ;SET LAST WORD FLAG
4090 020336 012737 000023 001216      MOV      #19.,WORK      ;SET UP TO TRANSFER LAST WORD
4091 020344 005337 001216          2$:      DEC      WORK          ;DONE YET?
4092 020350 001347          BNE      1$
4093
4094 020352 052737 000400 001170      BIS      #BIT8,ONCEE     ;SET TRANSFERRING CRC WORD
4095 020360 042737 000200 001170      BIC      #BIT7,ONCEE     ;CLEAR LAST WORD FLAG
4096
4097                                     ;GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
4098                                     ;EXC SHOULD THEN BE ASSERTED
4099
4100 020366 012777 177777 160520      MOV      #-1,RSER
4101 020374 004737 024430          JSR      PC,GENCRC      ;GENERATE CRC WORD
4102                                     ;AND LEAVE IN "WORK"
4103 020400 012702 030114          MOV      #INBUF,R2      ;GO TO END
4104 020404 062702 000200          ADD      #200,R2        ;OF DATA BUFFER
4105 020410 013712 001216          MOV      WORK,R2       ;LOAD CRC WORD
4106 020414 010205          MOV      R2,R5         ;RESET POINTER FOR
4107 020416 162705 000002          SUB      #2,R5         ;R5 FOR CRC WD
4108 020422 012737 000023 001216      MOV      #19.,WORK      ;SETUP TO XFER CRC
4109 020430 005337 001216          3$:      DEC      WORK          ;DONE YET?
4110 020434 001315          BNE      1$           ;NO

```



```

;NOW TEST CONTROLLER
4162
4163
4164 020560 022777 144260 160312 INTMR: CMP #144260, @RSCS1 : IS CS1 CORRECT?
4165 020566 001401 BEQ :+4 : YES
4166 020570 104014 HLT !DA!WC : YES
4167 020572 022777 000001 160310 5$: CMP #1, @RSDA : IS RSDA CORRECT?
4168 020600 001401 BEQ :+4 : YES
4169 020602 104004 HLT !DA : DA SHOULD = 1
4170 020604 022777 000004 160302 CMP #4, @RSER : DID RMR SET IN RSER
4171 020612 001401 BEQ :+4 : YES
4172 020614 104050 HLT !DS!WC : RSER SHOULD = 4
4173 020616 022777 000001 160264 CMP #1, @RSDA : DOES RSDA=1
4174 020624 001401 BEQ :+4 : YES
4175 020626 104004 HLT !DA : RSDA SHOULD=1
4176 020630 000240 INTMR1: NOP : DONE

```

K08

MAINDEC-11-DZRSE-C
DZRSEC.P11 TST66

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC
DISK ADDRESS OVERFLOW TEST

MACY11 27(732) 25-SEP-76 10:44 PAGE 102

```

4177 ;*****
4178 ;TEST 66 DISK ADDRESS OVERFLOW TEST
4179 ;*****
4180 020632 104400 †TST66: SCOPE
4181
4182 ;MODULES TESTED: M7754, M7771, M7770
4183 ;SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77
4184 ;TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER
4185 ;(LBT) BIT TO SET IN THE RSDS REGISTER.
4186
4187 020634 104414 MRAOE: CLRDK ;CLEAR ALL REGISTERS
4188 020636 012706 000500 MOV #500, SP ;SETUP STACK POINTER
4189 020642 104430 MRIND ;SEND INDEX PULSE TO MR REG
4190 020644 104420 MRCK ;CHECK MAINT REG
4191 020646 022701 22701 ;TO EQUAL 22701
4192 020650 104424 MRINT ;INITIALIZE MAINT REG BY SENDING
4193 ;2 CLOCK PULSES (CLEAR MRSP)
4194 020652 012777 007777 160230 MOV #7777, ARSDA ;SETUP DISK ADDRESS
4195 020660 012777 177400 160216 MOV #-400, ARSWC ;SETUP FOR A 2 SECTOR TRANSFER
4196 020666 012777 030714 160212 MOV #OUTBUF, ARSBA ;GET OUTPUT BUFFER
4197
4198 ;SETUP BUFFER WITH ALL ONES
4199 020674 012705 030714 MOV #OUTBUF, R5 ;GET STARTING ADDRESS OF OUTBUF
4200 020700 012737 000400 001204 MOV #400, REPT ;LOAD 2 SECTORS
4201 020706 012725 177777 1S: MOV #-1, (R5)+ ;WITH WORDS
4202 020712 005337 001204 DEC REPT ;OF ALL ONES
4203 020716 001373 BNE 1S
4204
4205 020720 012777 000061 160152 MOV #61, ARSCS1 ;LOAD WRITE COMMAND
4206 020726 104430 MRIND ;SET INDEX PULSE
4207
4208 ;SUPPLY CLOCKS TO STEP THROUGH A TRACK
4209
4210 020730 012737 000002 001204 MOV #2, REPT
4211 020736 012704 124000 5S: MOV #43008, R4 ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
4212 020742 012702 000011 MOV #11, R2
4213 020746 012703 000001 MOV #1, R3
4214 020752 010277 160146 2S: MOV R2, ARSMR
4215 020756 010377 160142 MOV R3, ARSMR
4216 020762 005304 DEC R4
4217 020764 001372 BNE 2S
4218 020766 005337 001204 DEC REPT
4219 020772 001361 BNE 5S
4220
4221 020774 104422 MRCLK ;CLOCK A 11 AND A 1 INTO RSMR
4222 020776 104426 DSCK ;CHECK MR
4223 021000 012400 12400 ;TO EQUAL 12400
4224 021002 104000 HLT ;LBT SHOULD BE SET IN RSDS

```

```

4225 ;ASSERT MAINTENANCE INDEX PULSE TO RESET DRIVE
4226 ;FOR THE SECOND REVOLUTION
4227
4228 021004 104430 MRIND ;ASSERT MAINT INDEX PULSE
4229 021006 005037 001176 CLR MCCNT ;CLEAR THE CLOCK COUNTER
4230 021012 104420 MRCK ;CHECK MR REG
4231 021014 002501 2501 ;TO EQUAL 2501. SHOULD STILL BE WRITING
4232 021016 104000 HLT
4233
4234 ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS03 RESYNC PERIOD
4235 021020 012737 001000 001204 MOV #512.,REPT ;CLOCK COUNT TO STEP THRU RESYNC
4236 021026 104446 4$: MCLK1 ;2ND REVOLUTION
4237 021030 104420 MRCK ;CHECK MR
4238 021032 012511 12511 ;TO EQUAL 12511
4239 021034 104000 HLT ;MR=BAD GOOD=CORRECT ANS
4240 021036 104450 MCLK0 ;CLOCK MR REG
4241 021040 104420 MRCK ;CHECK MR
4242 021042 002501 2501 ;REG TO
4243 021044 104000 HLT ;EQUAL 2501
4244 021046 005337 001204 DEC REPT
4245 021052 001365 BNE 4$ ;LOOP TILL DONE
4246
4247 ;SUPPLY 2 CLOCKS TO CAUSE THE SECTOR PULSE TO APPEAR IN
4248 ;THE MR REGISTER AND THE "AOE" ERROR TO APPEAR IN
4249 ;THE RSER REGISTER
4250
4251 021054 104422 AOECK: MRCLK
4252 021056 104422 MRCLK ;CAUSE SECTOR PULSE AND AOE ERROR
4253 021060 104420 MRCK ;CHECK FOR SECTOR PULSE
4254 021062 022701 22701 ;IN RSMR
4255 021064 104000 HLT ;MR=BAD GOOD=CORRECT ANS
4256 021066 022777 001000 160020 CMP #1000,RSER ;DID AOE SET IN RSER?
4257 021074 001401 BEQ 1$ ;AOE SHOULD BE SET IN RSER
4258 021076 104040 HLT !DS ;RSER SHOULD EQUAL 1000
4259 021100 022777 152600 160004 1$: CMP #152600,RSDS ;IS RSDS CORRECT
4260 021106 001401 BEQ 2$ ;YES
4261 021110 104040 HLT !DS ;ERR & ATA SHOULD BE SET IN RSDS
4262 ;BECAUSE OF AOE ERROR IN RSER
4263 021112 104414 2$: CLRDK ;CLEAR ERROR
4264 021114 005777 157774 TST RSER ;DID ERROR CLEAR?
4265 021120 001401 BEQ 3$ ;YES
4266 021122 104040 HLT !DS ;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
4267 021124 022777 010600 157760 3$: CMP #10600,RSDS ;DID ERRORS CLEAR
4268 021132 001401 BEQ +4 ;YES
4269 021134 104040 HLT !DS ;ERR AND ATA & LBT SHOULD ALL BE CLEARED
4270 ;FOR CLR WAS SET IN RSCS2

```

```

4271 ;MAINTENANCE MODE VERIFY TEST
4272 ;-----DANGER---THIS TEST DESTROYS DATA ON DISKS--DANGER
4273 ;THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH
4274 ;REGISTER" FOR IT WILL ACTUALLY WRITE DATA ONTO THE DISK. IT
4275 ;WILL WRITE ONE TRACK OF ALL ONES. THE PROGRAM THEN GOES BACK
4276 ;TO THE MAINT WRITE TEST AND WRITES ONE SECTOR OF DATA (ZERO'S, ONES, FLOATING
4277 ;ONES AND FILLS THE REMAINDER OF SECTOR WITH A PATTERN OF 146314)
4278 ;THE DRIVE IS THEN TAKEN OUT OF
4279 ;"MAINTENANCE MODE" AND THE TRACK IS THEN READ. THE TRACK
4280 ;SHOULD CONTAIN ALL ONES.
4281
4282 ;*****
4283 ;TEST 67 MAINTENANCE MODE VERIFY TEST
4284 ;*****
4285 021136 104400 TST67: SCOPE
4286
4287 ;MODULE TESTED G182
4288
4289 021140 032777 004000 157660 MRVR: BIT #BIT11,ASWR ;DO THIS TEST?
4290 021146 001002 BNE 3$ ;YES
4291 021150 000137 021660 JMP @#INFTST ;NO
4292 021154 012737 001600 001144 3$: MOV #1600,FLAG2 ;SET VERIFY TEST FLAG
4293 021162 104414 CLRDK ;CLEAR ALL DRIVES
4294 021164 012737 017777 001232 MOV #17777,WORKS ;STALL TO RESYNC
4295 021172 005337 001232 4$: DEC WORKS ;DRIVE
4296 021176 001375 BNE 4$ ;TIMING LOGIC
4297
4298 ;STEP THRU RESYNC PERIOD
4299
4300 021200 012777 170000 157676 MOV #-10000,@R5WC ;WRITE ONE TRACK - 4K WORDS
4301 021206 012737 177777 030114 MOV #177777,INBUF ;WRITE A PATTERN 12525
4302 021214 052777 000010 157660 BIS #BIT3,@R5CS2 ;SET BAI BIT
4303 021222 012777 030114 157656 MOV #INBUF,@R5BA ;SET DATA WD
4304 021230 012737 007777 001204 MOV #7777,REPT ;SETUP WAIT LOOP
4305 021236 012777 000061 157634 MOV #61,@R5CS1 ;GO WRITE
4306 021244 105777 157630 1$: TSTB @R5CS1 ;DONE YET?
4307 021250 100404 BMI 2$ ;YES
4308 021252 005337 001204 DEC REPT ;DECREMENT COUNTER WAITING
4309 021256 001372 BNE 1$ ;FOR READY
4310 021260 104000 HLT ;READY NEVER CAME UP
4311 021262 005777 157612 2$: TST @R5CS1 ;ANY ERRORS?
4312 021266 100002 BPL MRVR1 ;NO
4313 021270 104050 HLT ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
4314 021272 000433 BR TBDIA ;TYPE MESSAGE

```



```

4315 021274 104414 MRVRI: CLRDK ;CLEAR ALL REGISTERS
4316 021276 012777 170000 157600 MOV #-10000,ARSWC ;SETUP WC
4317 021304 052777 000010 157570 BIS #BIT3,ARSCS2 ;SET BAI
4318 021312 012777 030114 157566 MOV #INBUF,ARSB A ;SETUP RSBA
4319 021320 012737 007777 001204 MOV #7777,REPT ;SETUP WAIT LOOP
4320 021326 012777 000051 157544 MOV #51,ARSCS1 ;DO A WRITE CHECK TO VERIFY DISK
4321 021334 105777 157540 15: TSTB ARSCS1 ;TEST
4322 021340 100404 BMI 2$ ;FOR READY TO COME BACK
4323 021342 005337 001204 DEC REPT ;WAIT
4324 021346 001372 BNE 1$
4325 021350 104000 HLT ;READY NEVER CAME BACK
4326 021352 005777 157522 2$: TST ARSCS1 ;ANY ERRORS?
4327 021356 100032 BPL MRVRR ;NO
4328 021360 104050 HLT !DS!WC ;STOP HERE WC FAILED
4329 ;GO TO DZRSB DIAG
4330 ;BEFORE TRYING TO DEBUG
4331 ;THIS TEST
4332 021362 TBDIA:
4333 021362 104402 021366 TYPE .+2 ;.ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
4334 021444 000137 012320 MRVRR: JMP #MRWRT ;GO WRITE IN MAINTENANCE MODE
4335 ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
4336
4337 021450 104414 MRVR2: CLRDK ;CLEAR ALL REGISTERS
4338 021452 012737 007777 001216 MOV #7777,WORK ;STALL
4339 021460 005337 001216 3$: DEC WORK ;WAITING FOR
4340 021464 001375 BNE 3$ ;DRIVE TO GET IN SYNC WITH INDEX PULSE
4341 021466 012777 170000 157410 MOV #-10000,ARSWC ;SETUP WC FOR 1 TRACK
4342 021474 052777 000010 157400 BIS #BAI,ARSCS2 ;SET BAI
4343 021502 012777 030114 157376 MOV #INBUF,ARSB A ;SETUP RSBA
4344 021510 012737 177777 030114 MOV #177777,INBUF ;SETUP FOR COMPARE
4345 021516 012777 000051 157354 MOV #51,ARSCS1 ;DO A WRITE CHECK
4346 021524 105777 157350 15: TSTB ARSCS1 ;TEST FOR
4347 021530 100375 BPL 1$ ;READY TO COME BACK
4348 021532 032777 040000 157342 BIT #WCE,ARSCS2 ;DID WCE SET?
4349 021540 001442 BEQ 2$ ;NO
4350 021542 104402 021546 TYPE .+2 ;.ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
4351 021642 104040 HLT !DS
4352 021644 000404 BR 4$ ;GET OUT
4353 021646 005777 157226 2$: TST ARSCS1 ;ANY ERRORS?
4354 021652 100001 BPL .+4 ;NO
4355 021654 104040 HLT !DS ;SHOULD NOT HAVE ANY ERRORS HERE
4356 021656 000240 4$: NOP
4357 ;TRY DZRSB DIAGNOSTIC
4358
4359 021660 052737 000001 001170 INFTST: BIS #BIT0,ONCEE ;SET FOUND DRIVE FLAG
4360 021666 000137 002352 JMP #TRYNX ;GET NEXT DRIVE

```

```

4361          .SBTTL          $DONE - BELL AND SCOPE ROUTINE
4362
4363 021672 104400          DONE:  SCOPE          ; TERMINATING SCOPE FOR LOOPING
4364 021674 062737 000001 001006      ADD      #1,PCNT+2      ; ADD 1 TO THE PASS COUNT
4365 021702 005537 001004          ADC      PCNT          ; MAKE IT DOUBLE PREC.
4366 021706 032777 002000 157112      BIT      #SW10,JSWR    ; RING THE BELL?
4367 021714 001004          BNE      4$            ; NO!
4368 021716 104402 021722          TYPE     .+2          ; .ASCIZ <BELL><177>
4369 021726 013700 000042          4$:  MOV     #42,R0     ; GET MONITOR ADDRESS
4370 021732 001405          BEQ     $END1         ; IF NONE
4371 021734 000005          RESET
4372 021736 004710          SENDAD: JSR     7,(0)  ; GO TO MONITOR
4373 021740 000240 000240 000240      240,240,240          ; SAVE ROOM FOR ACT11
4374 021744 000137 021754          SEND1: JMP     MULSYS ; RETURN
4375
4376 021752 000000          .TBIT:  0            ; T BIT FLAG
4377
4378          ;MULTI DRIVE SYSTEM?
4379
4380          MULSYS:
4381 021754 104402 021760          TYPE     .+2          ; .ASCIZ <15><12>"END OF PASS"
4382 021776 005037 001010          CLR     (AO
4383 022002 005037 001000          CLR     ICNT
4384 022006 032737 000020 001146      BIT      @BIT4,FLAG3 ; MULTI DRIVE?
4385 022014 001002          BNE     1$            ; NO
4386 022016 000137 002006          JMP     @MULTI1       ; YES
4387 022022 000137 002564          1$:  JMP     @NOWGO    ; TEST ONLY ONE DRIVE
4388
4389          ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
4390
4391 022026 032737 000004 001170      NOPERR: BIT      @BIT2,ONCEE ; WERE WE HERE BEFORE?
4392 022034 001031          BNE     1$            ; YES
4393 022036 052737 000004 001170      BIS     @BIT2,ONCEE   ; SET BEEN HERE BEFORE FLAG
4394 022044 104402 022050          TYPE     .+2          ; .ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
4395 022112 013746 001216          MOV     WORK,-(6)    ; PUT WORK ON STACK
4396 022116 104406          TYPES
4397 022120 000207          1$:  RTS     PC

```

```

4398 022122 104402 022136          CHG:   TYPE   ,REGCHG       ;TYPE MESSAGE
4399 022126 013746 001216          MOV   WORK,-(6)  ;PUT WORK ON STACK
4400 022132 104406          TYPES          ;TYPE STACK IN OCTAL - SUPRESS
4401 022134 000207          RTS   PC
4402
4403 022136 044103 047101 042507  REGCHG: .ASCIZ  "CHANGED WITH NO-OP FUNCTION "
4404 022144 020104 044527 044124
4405 022152 047040 026517 050117
4406 022160 043040 047125 052103
4407 022166 047511 020116      000
4408
4409 022173      015 051012 051115 TRMR:   .ASCIZ  '<15><12>"RMR DID NOT SET BY WRITING INTO "'
4410 022200 020040 044504 020104
4411 022206 047516 020124 042523
4412 022214 020124 054502 053440
4413 022222 044522 044524 043516
4414 022230 044440 052116 020117
4415 022236      000
4416      022240          .EVEN
4417
4418 022240 104422          .MRINT: MRCLK          ;CLOCK THE MAINT REG WITH A 11 AND A 1
4419 022242 104422          MRCLK          ;SAME
4420 022244 000002          RTI          ;RETURN
4421
4422 022246 012777 000011 156650 .MRCLK: MOV   #11,RSMR    ;CLOCK THE
4423 012777 000001 156642          MOV   #1,RSMR        ;MAINT REG
4424 0F2737 000001 001200          ADD   #1,MCCNT+2     ;ADD 1 TO CLOCK COUNT
4425 0L537 001176          ADC   MCCNT          ;MAKE DOUBLE PRECISION
4426 022274 000002          RTI
4427
4428 022276 017700 156622          .MRCK: MOV   @RSMR,BAD   ;GET THE CONTENTS OF RSMR
4429 017601 000000          MOV   @2(SP),GOOD      ;GET THE CORRECT ANSWER
4430 062716 000002          ADD   #2,(SP)          ;UPDATE THE RETURN ADDRESS FOR AN ERROR
4431 022312 020100          CMP   GOOD,BAD        ;IS THE MR REG CORRECT?
4432 022314 001002          BNE   IS              ;NO EXIT
4433 0L2716 0F2716 000002          ADD   #2,(SP)          ;UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4434 022322 000002          IS:   RTI          ;RETURN
4435
4436          ;SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4437 022324 012777 000021 156572 .MRIND: MOV   #21,RSMR    ;SEND INDEX
4438 022332 012777 000001 156564          MOV   #1,RSMR        ;PULSE TO MR REG
4439 022340 000002          RTI
4440
4441 017342 017700 156544          .DSCK: MOV   @RSDS,BAD   ;GET THE CONTENTS OF RSDS
4442 0L346 017601 000000          MOV   @2(SP),GOOD      ;GET THE CORRECT ANS
4443 0L52 062716 000002          ADD   #2,(SP)          ;UPDATE THE RETURN ADDR FOR AN ERROR
4444 022356 020100          CMP   GOOD,BAD        ;IS RSDS CORRECT
4445 022360 001002          BNE   IS              ;NO EXIT
4446 022362 062716 000002          ADD   #2,(SP)          ;UPDATE RETURN ADDR TO SKIP THE HLT FOR CORRECT ANS
4447 022366 000002          IS:   RTI

```

```

;GET 1 BIT OF DATA FROM BUFFER
;SAVE THE LAST BIT TRANSFERED IN LOCATION LSTOD
447
448
449
450 022370 032737 000200 001144 .XBIT: BIT #BIT7,FLAG2 ;1ST 1 BIT OF 1ST WD?
451 022376 001446 BEQ 2S ;NO
452 022400 012737 000001 001152 MOV #1,LSTOD ;YES SETUP SYNC 1 BIT FOR END OF PREAMBLE;
;TO CALCULATE BOTTOM BIT
453 022406 032737 000100 001170 BIT #BIT6,ONCEE ;1ST TIME THROUGH?
454 022414 001006 BNE 5S ;YES
455 022416 042737 000200 001144 BIC #BIT7,FLAG2
456 022424 012737 000000 001152 MOV #0,LSTOD
457 022432 042737 000100 001170 5S: BIC #BIT6,ONCEE ;CLEAR 1ST TIME THROUGH FLAG
458 022440 005037 001210 4S: CLR CLKCNT ;CLEAR CLOCK COUNTER AT START OF EACH WD
459 022444 C 737 000400 001170 BIT #BIT8,ONCEE ;ON CRC WD?
460 022452 001062 BNE 1S ;YES
461 022454 005037 001156 CLR NOWOD ;NO BITS 16 & 17 ARE 0
462
463
464 022460 032737 000400 001144 BIT #BIT8,FLAG2 ;XFERING BIT 17?
465 022466 001003 BNE 7S ;YES
466 022470 042737 001000 001144 BIC #BIT9,FLAG2 ;CLEAR FLAG FOR BIT 16
467 022476 042737 000400 001144 7S: BIC #BIT8,FLAG2 ;CLEAR FLAG FOR BIT 17
468 022504 012737 000020 001226 6S: MOV #16.,WORK3 ;LOOP 16 TIMES 1 FOR EACH BIT
469 022512 000002 RTI ;EXIT
470 022514 013737 001156 001152 2S: MOV NOWOD,LSTOD ;SAVE LAST BIT XFERED
471 022522 032737 001000 001144 BIT #BIT9,FLAG2
472 022530 001343 BNE 4S
473 022532 005737 001226 TST WORK3 ;DONE WITH WD YET?
474 022536 001013 BNE 3S ;NO
475 022540 032737 002000 001144 BIT #BIT10,FLAG2 ;ON BIT 16 OF CRC WD?
476 022546 001334 BNE 4S ;YES
477 022550 062705 000032 ADD #2,R5 ;UPDATE BUFFER WD
478 022554 011504 MOV (R5),R4 ;GET DATA WD
479 022556 052737 001400 001144 BIS #1400,FLAG2 ;SET BITS 8 & 9 IN FLAG2
480 022564 000725 BR 4S
481 022566 005037 001156 3S: CLR NOWOD ;CLEAR PRESENT BIT
482 022572 032737 001000 001144 BIT #BIT9,FLAG2 ;DID WE XFER BITS 16 & 17 YET?
483 022580 001317 BNE 4S ;NO
484 022602 000241 CLC
485 022604 006104 ROL R4 ;GET NEXT DATA BIT
486 022606 006137 001156 ROL NOWOD ;PUT IT INTO NOWOD
487 022612 005337 001226 DEC WORK3 ;KEEP COUNT OF BITS IN THE WORD
488 022616 000002 RTI ;EXIT
489
490 ;CRC IS BEING WRITTEN. BITS 17 & 16 ARE DATA BITS
491 ;BITS 0 & 1 ARE ALWAYS 0
492
493 022620 005037 001156 1S: CLR NOWOD ;CLEAR PRESENT BIT
494 022624 006104 ROL R4 ;GET NEXT BIT
495 022626 006137 001156 ROL NOWOD ;TO BE XFERED
496 022632 032737 002000 001144 BIT #BIT10,FLAG2 ;DONE WITH BITS 16 & 17 YET?
497 022640 C 1321 BNE 6S ;YES
498 022642 0 737 002000 001144 BIS #BIT10,FLAG2 ;NO
499 022650 042737 001000 001144 BIC #BIT9,FLAG2
500 022656 000002 RTI ;EXIT

```

```

4501                                     ;CLOCK ROUTINE (1ST OF ONE) WHICH IS USED TO CLOCK ONE BIT OF
4502                                     ;DATA TO THE DRIVE AT A TIME. THIS ROUTINE ALSO CHECKS THE PREVIOUS
4503                                     ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
4504                                     ;THE MWDB BIT (BIT 12 IN THE MR REG) SHOULD BE IN
4505
4506
4507 022660 104446 .CLKD1: MCLKI                                     ;CLOCK MR REG WITH AN 11
4508 022662 005003 CLR R3                                       ;CLEAR WORK LOCATION
4509 022664 005737 001156 TST NOWOD                               ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
4510 022670 001005 BNE TSTEV B                                     ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
4511                                     ;NOW BIT IS A 1 MWDB IS 0
4512 022672 005737 001152 1$: TST LSTOD                               ;TEST THE LAST ODD DATA BIT THAT WAS SENT
4513 022676 001002 BNE TSTEV B                                     ;LAST ODD DATA BIT WAS A 1
4514                                     ;MWDB IS A 0
4515 022700 052703 010000 2$: BIS #BIT12,R3                       ;SET MWDB FOR LATER COMPARE WITH MR REG
4516
4517 022704 012701 123511 TSTEV B: MOV #123511,G00D                ;GET CORRECT ANS
4518 022710 050301 BIS R3,G00D                                       ;FOR MR REG
4519 022712 004737 024224 JSR PC,MRCAL                               ;DETERMINE STATE OF SB & LSR BITS
4520 022716 017700 156202 MOV @R5MR,BAD                          ;GET CONTENTS OF MR REG
4521 022722 020100 CMP GOOD,BAD                                       ;IS MR REG CORRECT?
4522 022724 001002 BNE 2$                                           ;NO TYPE OUT MR REG
4523 022726 062716 000002 ADD #2,(SP)                               ;UPDATE RETURN ADDR FOR CORRECT ANS
4524 022732 000002 2$: RTI                                           ;RETURN

```

```

4525          :SECOND CLOCK ROUTINE WHICH WILL FINISH TRANSFERRING THE DATA BIT
4526          :THIS ROUTINE WILL CALCULATE WHAT MWDB SHOULD EQUAL IN THE
4527          :MAINTENANCE REGISTER
4528
4529 022734 104450 .CLKDD: MCLKD          :CLOCK MR REG
4530 022736 005737 001156      TST      NOWOD          :IS THE PRESENT DATA BIT A 1?
4531 022742 001403          BEQ      1$              :NO IT IS A 0
4532 022744 052703 010000      BIS      #BIT12,R3      :SET MWDB FOR BIT BEING SENT IS A 1
4533 022750 000402          BR      4$
4534 022752 042703 010000 1$:  BIC      #BIT12,R3      :CLEAR MWDB FOR PRESENT BIT IS A 0
4535 022756 012701 023501 4$:  MOV      #23501,GOOD    :GET CORRECT ANS
4536 022762 050301          BIS      R3,GOOD        :FOR MR REG
4537 022764 004737 024224      JSR      PC,MRCAL       :DETERMINE STATE OF SB & LSR BITS
4538 022770 017700 156130      MOV      @R5MR,BAD     :GET CONTENTS OF MR REG
4539 022774 020100          CMP      GOOD,BAD     :IS MR REG CORRECT?
4540 022776 001002          BNE      5$          :NO TYPEOUT ERROR
4541 023000 062716 000002      ADD      #2,(SP)      :UPDATE ADDR FOR CORRECT ANS
4542 023004 000002 5$:      RTI          :RETURN
4543
4544          :TYPEOUT ROUTINE TO DETERMINE WHICH IC FAILED IN CRC TEST2
4545          :AND TO TYPE IT OUT
4546
4547 023006 012737 023116 001216 CRCTYP: MOV      #CRCTAB,WORK    :GET STARTING LOC OF IC TABLE
4548 023014 012737 000001 001222      MOV      #1,WORK1     :SETUP TO TEST FIRST CHIP
4549 023022 033737 001222 001174 1$:  BIT      WORK1,SAVEE  :WAS IT THIS BIT?
4550 023030 001006          BNE      2$          :YES TYPE IT
4551 023032 062737 000006 001216      ADD      #6,WORK      :NO INDEX TABLE POINTER
4552 023040 006137 001222          ROL      WORK1        :SETUP TO TEST NEXT CHIP
4553 023044 000766          BR      1$          :NOW TES IT
4554 023046 004777 156144 2$:  JSR      PC,@WORK     :TYPE OUT CHIP
4555 023052 104402 023056          TYPE      #2
4556 023114 000207          RTS      PC

```

;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE

4557						
4558						
4559	023116	104402	023306	CRCTAB:	TYPE	E302
4560	023122	000207			RTS	PC
4561	023124	104402	023314		TYPE	E305
4562	023130	000207			RTS	PC
4563	023132	104402	023322		TYPE	E307
4564	023136	000207			RTS	PC
4565	023140	104402	023330		TYPE	E3010
4566	023144	000207			RTS	PC
4567	023146	104402	023337		TYPE	E3012
4568	023152	000207			RTS	PC
4569	023154	104402	023346		TYPE	E3015
4570	023160	000207			RTS	PC
4571	023162	104402	023355		TYPE	E242
4572	023166	000207			RTS	PC
4573	023170	104402	023363		TYPE	E245
4574	023174	000207			RTS	PC
4575	023176	104402	023371		TYPE	E247
4576	023202	000207			RTS	PC
4577	023204	104402	023377		TYPE	E2410
4578	023210	000207			RTS	PC
4579	023212	104402	023406		TYPE	E2412
4580	023216	000207			RTS	PC
4581	023220	104402	023415		TYPE	E2415
4582	023224	000207			RTS	PC
4583	023226	104402	023424		TYPE	E192
4584	023232	000207			RTS	PC
4585	023234	104402	023432		TYPE	E197
4586	023240	000207			RTS	PC
4587	023242	104402	023440		TYPE	E1910
4588	023246	000207			RTS	PC
4589	023250	104402	023447		TYPE	E1915
4590	023254	000207			RTS	PC

;CLOCK MR REG WITH A 0-1

4591						
4592						
4593						
4594	023256	012777	000001	155640	.MCLKB:	MOV #1,DRSMR
4595	023264	012777	000011	155632		MOV #11,DRSMR
4596	023272	062737	000001	001200		ADD #1,MCCNT+2
4597	023300	005537	001176			ADC MCCNT
4598	023304	000002				RTI

4599	023306	031505	026460	000062	E302:	.ASCIZ	"E30-2"
4600	023314	031505	026460	000065	E305:	.ASCIZ	"E30-5"
4601	023322	031505	026460	000067	E307:	.ASCIZ	"E30-7"
4602	023330	031505	026460	030061	E3010:	.ASCIZ	"E30-10"
4603	023336	000					
4604	023337	105	030063	030455	E3012:	.ASCIZ	"E30-12"
4605	023344	000062					
4606	023346	031505	026460	032461	E3015:	.ASCIZ	"E30-15"
4607	023354	000					
4608	023355	105	032062	031055	E242:	.ASCIZ	"E24-2"
4609	023362	000					
4610	023363	105	032062	032455	E245:	.ASCIZ	"E24-5"
4611	023370	000					
4612	023371	105	032062	033455	E247:	.ASCIZ	"E24-7"
4613	023376	000					
4614	023377	105	032062	030455	E2410:	.ASCIZ	"E24-10"
4615	023404	000060					
4616	023406	031105	026464	031061	E2412:	.ASCIZ	"E24-12"
4617	023414	000					
4618	023415	105	032062	030455	E2415:	.ASCIZ	"E24-15"
4619	023422	000065					
4620	023424	030505	026471	000062	E192:	.ASCIZ	"E19-2"
4621	023432	030505	026471	000067	E197:	.ASCIZ	"E19-7"
4622	023440	030505	026471	030061	E1910:	.ASCIZ	"E19-10"
4623	023446	000					
4624	023447	105	034461	030455	E1915:	.ASCIZ	"E19-15"
4625	023454	000065					
4626							
4627							
4628							
4629	023456	012777	000011	155440	.MCLK1: MOV	#11, @RSMR	
4630	023464	062737	000001	001200	ADD	#1, MCCNT+2	
4631	023472	005537	001176		ADC	MCCNT	
4632	023476	000002			RTI		
4633							
4634							
4635							
4636	023500	012777	000001	155416	.MCLK0: MOV	#1, @RSMR	
4637	023506	003002			RTI		

;CLOCK MR REG WITH A 1

;CLOCK MR REG WITH A0


```

4638                                     ;GET ONE BIT OF DATA FROM INBUF
4639                                     ;FOR READING FROM DRIVE TO DETERMINE THE
4640                                     ;STATE OF MR0B IN THE MR REG.
4641
4642 023510 005737 001226 .RBIT: TST      WORK3      ;STARTING NEW WD?
4643 023514 001035          BNE      3$          ;NO
4644 023516 062705 000002 ADD      #2,R5      ;UPDATE BUFFER WD
4645 023522 011504          MOV      (R5),R4     ;GET DATA WD
4646 023524 052737 004000 001144 BIS      #BIT11,FLAG2 ;SET TO INDICATE BIT 17
4647 023532 005037 001210 5$: CLR      CLKCNT     ;CLEAR CLOCK COUNTER AT START OF EACH WD
4648 023536 032737 000400 001170 BIT      #BIT8,ONCE  ;ON CRC WD?
4649 023544 001041          BNE      1$          ;YES
4650 023546 032737 000040 001144 BIT      #BITS,FLAG2 ;IN CRC TEST ???
4651 023554 001407          BEQ      7$          ;NO
4652 023556 012737 000020 001226 MOV      #16,WORK3
4653 023564 042737 004000 001144 BIC      #BIT11,FLAG2 ;FOR CRC TEST
4654 023572 000416          BR       4$
4655 023574 005037 001156 7$: CLR      NOWOD     ;BITS 16 + 17 OF DATA WORD ARE 0
4656 023600 012737 000020 001226 6$: MOV      #16.,WORK3 ;16 LOOPS FOR REMAINING 16 BITS OF WORD
4657 023606 000002          RTI
4658 023610 032737 004000 001144 3$: BIT      #BIT11,FLAG2 ;IS THIS BIT 16?
4659 023616 001404          BEQ      4$          ;NO
4660 023620 042737 004000 001144 BIC      #BIT11,FLAG2 ;RTRANSFER BIT 16
4661 023626 000741          BR       5$
4662 023630 005037 001156 4$: CLR      NOWOD     ;CLEAR PRESENT BIT
4663 023634 006104          ROL      R4          ;GET NEXT DATA BIT
4664 023636 006137 001156 ROL      NOWOD     ;SAVE IT IN ODD BIT
4665 023642 005337 001226 DEC      WORK3      ;KEEP COUNT OF BITS IN THE WORD
4666 023646 000002          RTI          ;RETURN
4667                                     ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
4668 023650 005037 001156 1$: CLR      NOWOD     ;GET BITS 17
4669 023654 006104          ROL      R4          ;AND 16
4670 023656 006137 001156 ROL      NOWOD     ;FOR CRC WORD
4671 023662 000746          BR       6$          ;CONTINUE

```

4672	023664	004737	024202		.CLKR1:	JSR	PC,CALRTB	;CALCULATE MRDB BIT FOR MR REG
4673	023670	012703	000011			MOV	#11,R3	;SETUP CLOCK BITS
4674	023674	062737	000001	001200		ADD	#1,MCCNT+2	;INCREMENT
4675	023702	001537	001176			ADC	MCCNT	;CLOCK COUNT
4676	023706	053703	001216			BIS	WORK,R3	;SET BOTTOM BITS
4677	023712	010377	155206			MOV	R3,RSMR	;SEND
4678	023716	012701	133611			MOV	#133611,GOOD	;CALCULATE CORRECT ANS FOR MR REG
4679	023722	032737	000004	001144		BIT	#BIT2,FLAG2	;WRITE CK TEST?
4680	023730	001402				BEQ	7\$;NO
4681	023732	052701	000100			BIS	#BIT6,GOOD	;YES SET RD IN MR REG
4682	023736	050301			7\$:	BIS	R3,GOOD	
4683	023740	032737	000400	001170		BIT	#BIT8,ONCEE	;ON CRC WD?
4684	023746	001406				BEQ	2\$;NO
4685	023750	022737	000022	001204		CMP	#22,REPT	;SHOULD CRCW BE SET?
4686	023756	001402				BEQ	2\$;YES
4687	023760	042701	020000			BIC	#20000,GOOD	;CLEAR CRCW
4688	023764	005337	001206		2\$:	DEC	REPT1	;SHOULD SB SET
4689	023770	001017				BNE	6\$;NO
4690	023772	012737	000044	001206		MOV	#44,REPT1	;RESET SB COUNTER
4691	024000	052701	004000			BIS	#BIT11,GOOD	;SET SB
4692	024004	032737	000400	001170	3\$:	BIT	#BIT8,ONCEE	;ON CRC WD?
4693	024012	001406				BEQ	6\$;NO
4694	024014	022737	000043	001206		CMP	#43,REPT1	;SHOULD SB AND CRCW BE SET ?
4695	024022	001002				BNE	6\$;NO
4696	024024	052701	020000			BIS	#20000,GOOD	;SET SB AND CRCW
4697	024030	017700	155070		6\$:	MOV	RSMR,BAD	;GET MR REG
4698	024034	020100				CMP	GOOD,BAD	;IS RSMR CORRECT?
4699	024036	001002				BNE	4\$;NO
4700	024040	062716	000002			ADD	#2,(SP)	;YES
4701	024044	000002			4\$:	RTI		;RETURN

K09

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RSG3 MAINTENANCE MODE DIAGNOSTIC
\$DONE - BELL AND SCOPE ROUTINE

MACY11 27(732) 25-SEP-76 10:44 PAGE 115

4702	024046	053703	001216		.CLKRO:	BIS	WORK,R3		;SET BOTTOM BITS
4703	024052	042703	000010			BIC	#BIT3,R3		
4704	024056	010377	155042			MOV	R3,RSMR		;SEND
4705	024062	012701	023601			MOV	#23601,GOOD		;CALCULATE CORRECT ANS FOR MR REG
4706	024066	032737	000004	001144		BIT	#BIT2,FLAG2		;WRITE CK TEST?
4707	024074	001402				BEQ	7\$;NO
4708	024076	052701	000100			BIS	#BIT6,GOOD		;YES SET RD IN MR REG
4709	024102	050301			7\$:	BIS	R3,GOOD		
4710	024104	032737	000400	001170		BIT	#BIT8,ONCEE		;ON CRC WD?
4711	024112	001402				BEQ	2\$;NO
4712	024114	042701	020000			BIC	#20000,GOOD		;CLEAR CRCW
4713	024120	005337	001206		2\$:	DEC	REPT1		;SHOULD SB SET?
4714	024124	001017				BNE	6\$;NO
4715	024126	012737	000022	001206		MOV	#18,REPT1		;RESET SB COUNTER
4716	024134	052701	004000			BIS	#BIT11,GOOD		;SET SB
4717	024140	032737	000400	001170	3\$:	BIT	#BIT8,ONCEE		;ON CRC WD?
4718	024146	001406				BEQ	6\$;NO
4719	024150	022737	000022	001206		CMP	#22,REPT1		;SHOULD SB AND CRCW BE SET ?
4720	024156	001002				BNE	6\$;NO
4721	024160	052701	020000			BIS	#20000,GOOD		;SET SB AND CRCW
4722	024164	017700	154734		6\$:	MOV	RSMR,BAD		;GET MR REG
4723	024170	020100				CMP	GOOD,BAD		;IS RSMR CORRECT?
4724	024172	001002				BNE	4\$;NO
4725	024174	062716	000002			ADD	#2,(SP)		;YES
4726	024200	000002			4\$:	RTI			;RETURN

```

4727          ;CALCULATE THE STATE OF MRDB FROM CURRENT INPUT BIT
4728          ;LOCATION WORK CONTAINS CORRECT DATA FOR MRDB
4729 024202 005037 001216          CALRTB: CLR      WORK      ;CLEAR WORK LOCATION
4730 024206 005737 001156          TST      NOWOD     ;IS CURRENT BIT A 0?
4731 024212 001403                BEQ      2$        ;YES
4732 024214 052737 000004 001216  BIS      #BIT2,WORK ;NO SET MRDB
4733 024222 000207                RTS      PC          ;RETURN
4734
4735          ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
4736          ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR
4737
4738 024224 005237 001210          MRCAL: INC      CLKCNT   ;ADD ONE TO CLOCK COUNT OF WORD
4739 024230 032737 000200 001170  BIT      #BIT7,ONCEE ;TRANSFERRING LAST WORD?
4740 024236 001032                BNE      LSTWD     ;YES
4741 024240 032737 000400 001170  BIT      #BIT8,ONCEE ;TRANSFERRING CRC WORD?
4742 024246 001051                BNE      CRCWD     ;YES
4743 024250 022737 000016 001210  CMP      #16,CLKCNT ;CLOCK COUNT 16 OR GREATER?
4744 024256 101401                BLOS    1$        ;YES
4745 024260 000406                BR       2$        ;GET OUT
4746 024262 022737 000040 001210 1$: CMP      #40,CLKCNT ;CLOCK COUNT 40 OR GREATER?
4747 024270 101402                BLOS    2$        ;YES GET OUT
4748 024272 052701 004000          BIS      #BIT11,GOOD ;SET SB BIT
4749 024276 022737 000037 001210 2$: CMP      #37,CLKCNT
4750 024304 001404                BEQ      3$        ;
4751 024306 022737 000040 001210  CMP      #40,CLKCNT
4752 024314 001002                BNE      4$        ;
4753 024316 042701 002000          3$: BIC      #BIT10,GOOD ;CLEAR LSR
4754 024322 000207                4$: RTS      PC          ;RETURN
4755
4756          ;CALCULATE MR FOR LAST DATA WORD
4757 024324 022737 000036 001210 LSTWD: CMP      #36,CLKCNT ;IS THIS CLOCK 36 OR LESS?
4758 024332 103016                BHIS    2$        ;YES GETOUT
4759 024334 022737 000037 001210  CMP      #37,CLKCNT ;IS THIS CLOCK 15?
4760 024342 001003                BNE      3$        ;NO
4761 024344 042701 002000          4$: BIC      #BIT10,GOOD ;YES CLEAR LSR
4762 024350 000407                BR       2$        ;
4763 024352 022737 000040 001210 3$: CMP      #40,CLKCNT
4764 024360 001001                BNE      5$        ;
4765 024362 000770                BR       4$        ;
4766 024364 042701 020000          5$: BIC      #BIT13,GOOD ;CLEAR CRCW BIT
4767 024370 000207                2$: RTS      PC          ;
4768
4769          ;CALCULATE MR FOR CRC WORD
4770
4771 024372 042701 020000          CRCWD: BIC      #BIT13,GOOD ;CLEAR CRCW BIT
4772 024376 022737 000037 001210  CMP      #37,CLKCNT ;IS THIS CLOCK 17?
4773 024404 001002                BNE      2$        ;NO
4774 024406 042701 002000          BIC      #BIT10,GOOD ;CLEAR LSR BIT
4775 024412 022737 000040 001210 2$: CMP      #40,CLKCNT
4776 024420 001002                BNE      1$        ;
4777 024422 042701 002000          BIC      #BIT10,GOOD
4778 024426 000207                1$: RTS      PC          ;RETURN

```

```

4779
4780           ; GENERATE A CRC WORD FROM THE DATA BUFFER
4781           ; AND LEAVE THE CRC WORD IN "WORK" LOCATION
4782           ; EXIT ROUTINE WITH RTS PC
4783
4784 024430 012737 000100 001204 GENCRC: MOV      #64.,REPT      ; 64 WORDS PER SECTOR
4785 024436 032737 000040 001144           BIT      #BITS,FLAG2 ; IN CRC TEST?
4786 024444 001403           BEQ      13$           ; NO
4787 024446 012737 000110 001204           MOV      #72.,REPT      ; YES
4788 024454 012705 030114           13$: MOV      #INBUF,R5    ; GET STARTING ADDR OF OUTPUT BUFFER
4789 024460 011504           MOV      (R5),R4        ; GET DATA WD
4790 024462 005037 001220           CLR      WORK0         ; CLEAR WORK LOCATION
4791
4792           ; INBIT CONTAINS PRESENT INPUT BIT
4793           ; WK15 = BIT15 IF CRC AT TIME T
4794           ; WORK0 = CRC AT TIME T + DURING FINAL MANIPULATION
4795           ; WORK = BITS FROM SAVED CRC WORD (WCRC)
4796
4797 024466 012737 000022 001206 1$: MOV      #18.,REPT1   ; GET 18 BITS PER WD
4798 024474 032737 000040 001144           BIT      #BITS,FLAG2 ; IN CRC TEST?
4799 024502 001403           BEQ      2$           ; NO
4800 024504 012737 000020 001206           MOV      #16.,REPT1   ; YES
4801 024512 013737 001220 001202 2$: MOV      WORK0,WCRC  ; SAVE CURRENT CRC WD
4802 024520 005037 001214           CLR      WK15         ; CLEAR BIT 15 FROM CRC AT T 1
4803 024524 000241           CLC                    ; CLEAR CARRY
4804 024526 006137 001220           ROL      WORK0        ; SHIFT CRC WD LEFT
4805 024532 006137 001214           ROL      WK15         ; CONTAINS BIT 15 OF CRC
4806 024536 032737 000040 001144           BIT      #BITS,FLAG2 ; IN CRC TEST?
4807 024544 001004           BNE      12$          ; YES
4808 024546 022737 000021 001206           CMP      #17.,REPT1   ; DONE BITS 16 AND 17 YET?
4809 024554 101406           BLOS    3$           ; NO
4810 024556 005037 001212           12$: CLR      INBIT      ; CLEAR WORK LOC
4811 024562 006104           ROL      R4           ; PUT DATA BIT FROM BUFFER
4812 024564 006137 001212           ROL      INBIT        ; IN WORK1 LOC
4813 024570 000402           BR      4$           ;
4814 024572 005037 001212           3$: CLR      INBIT      ; FOR BITS 16 AND 17
4815 024576 013737 001214 001216 4$: MOV      WK15,WORK    ; GET BIT 15 OF CRC
4816 024604 004737 025030 001220 5$: JSR      PC,XXOR      ; XOR BIT15 WITH INPUT BIT
4817 024610 042737 000001 001220           BIC      #BIT0,WORK0  ;
4818 024616 005737 001212           TST      INBIT        ; TEST RESULT OF XOR
4819 024622 001403           BEQ      6$           ;
4820 024624 052737 000001 001220           BIS      #BIT0,WORK0  ;
4821 024632 013737 001212 001160 6$: MOV      INBIT,R50    ; SAVE XOR RESULT OF BIT 0 AND INPUT

```

```

4822                ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CLACULATE
4823                ;NEW B2 FOR WORK0
4824
4825 024640 005037 001216          CLR      WORK
4826 024644 032737 000002 001202  BIT      #BIT1,WCRC
4827 024652 001403                BEQ      7$
4828 024654 052737 000001 001216  BIS      #BIT0,WORK
4829 024662 013737 001160 001212 7$:  MOV      RSO,INBIT
4830 024670 004737 025030          JSR      PC,XXOR
4831 024674 042737 000004 001220  BIC      #BIT2,WORK0
4832 024702 005737 001212          TST      INBIT          ;TEST RESULT OF XOR
4833 024706 001403                BEQ      8$
4834 024710 052737 000004 001220  BIS      #BIT2,WORK0
4835
4836                ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
4837
4838 024716 005037 001216          8$:  CLR      WORK
4839 024722 032737 040000 001202  BIT      #BIT14,WCRC
4840 024730 001403                BEQ      9$
4841 024732 052737 000001 001216  BIS      #BIT0,WORK
4842 024740 013737 001160 001212 9$:  MOV      RSO,INBIT
4843 024746 004737 025030          JSR      PC,XXOR
4844 024752 042737 100000 001220  BIC      #BIT15,WORK0
4845 024760 005737 001212          TST      INBIT          ;TEST RESULT OF XOR
4846 024764 001403                BEQ      10$
4847 024766 052737 100000 001220  BIS      #BIT15,WORK0
4848 024774 005337 001206          10$: DEC      REPT1          ;DONE WITH WD
4849 025000 001244                BNE      2$              ;NO
4850 025002 005337 001204          DEC      REPT          ;DONE WITH SECTOR?
4851 025006 001404                BEQ      11$            ;YES
4852 025010 062705 000002          ADD      #2,R5          ;GET NEXT WD
4853 025014 011504                MOV      (R5),R4        ;GET DATA WD
4854 025016 000623                BR       1$
4855 025020 013737 001220 001216 11$: MOV      WORK0,WORK    ;SAVE CRC WORD IN WORK
4856 025026 000207                RTS      PC            ;EXIT
4857
4858                ;XOR SUBROUTINE
4859
4860 025030 013703 001216          XXOR: MOV      WORK,R3
4861 025034 043703 001212          BIC      INBIT,R3
4862 025040 043737 001216 001212  BIC      WORK,INBIT
4863 025046 050337 001212          BIS      R3,INBIT
4864 025052 000207                RTS      PC

```

.SBTTL \$TYPE - TTY TYPEOUT ROUTINE

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

4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875 025054 010446
4876 025056 010546
4877 025060 017605 000004
4878 025064 032705 177400
4879 025070 001002
4880 025072 016605 000004
4881 025076 105715
4882 025100 001423
4883 025102 122715 000012
4884 025106 001012
4885 025110 113704 001015
4886 025114 113777 001014 153702 55:
4887 025122 105777 153670
4888 025126 100375
4889 025130 005304
4890 025132 001370
4891 025134 112577 153664 45:
4892 025140 105777 153652
4893 025144 100375
4894 025146 000753
4895 025150 017646 000004 25:
4896 025154 062766 000002 000006
4897 025162 022666 000004
4898 025166 001006
4899 025170 062705 000002
4900 025174 042705 000001
4901 025200 010566 000004
4902 025204 012605 35:
4903 025206 012604
4904 025210 000002

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

```

4905          .SBTTL          $SCOPE - SCOPE LOOP HANDLER
4906
4907          :THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
4908          :LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
4909          :"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
4910          :RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
4911
4912          .SCOPE: KBDIN          ;GO CHECK FOR IG
4913          BIT          @SW8,@SWR          ;LOOP ON SPEC. TEST?
4914          BEQ          IS          ;NO LOOP ON SPEC. TEST
4915          CMPB          @SWR,ICNT          ;ON RIGHT TEST? *SW7-0*
4916          BEQ          .OVER          ;NOT RIGHT TEST
4917          BIT          @SW14,@SWR          ;LOOP ON TEST?
4918          BNE          .KIT          ;LOOP ON TEST IS SET
4919          BR          3$          ;SKIP - NOP FOR XOR TESTER
4920          MOV          @#4,-(6)          ;PUSH @#4 ON STACK
4921          MOV          @#4,@#4          ;SET FOR TIMEOUT
4922          TST          @#177060          ;ERROR ON XOR?
4923          MOV          (6)+,@#4          ;POP STACK INTO @#4
4924          BR          .SVLAD          ;NO ERROR - GO TO NEXT TEST
4925          CMP          (6)+,(6)+          ;CLEAR STACK
4926          MOV          (6)+,@#4          ;POP STACK INTO @#4
4927          BR          .KIT          ;ERROR - LOOP ON TEST
4928          BIT          @SW11,@SWR          ;KILL ITERATIONS
4929          BNE          .SVLAD          ;YES - KILL ITERATIONS
4930          TSTB          ICNT+1          ;FIRST ONE?
4931          BEQ          2$          ;BRANCH IF FIRST
4932          CMPB          TIMES,ICNT+1          ;DONE?
4933          BGT          .KIT          ;BRANCH IF NOT
4934          MOVB          @1,ICNT+1          ;FIRST ITERATION
4935          .SVLAD: INCB          ;COUNT TEST NUMBERS
4936          MOV          (6),LAD          ;SAVE LOOP ADDRESS
4937          MOV          ICNT,@DISPLAY          ;DISPLAY TEST NO. AND ITERATION COUNT
4938          RTI          ;RETURN
4939
4940          .KIT: INCB          ICNT+1          ;INC THE ITERATION COUNT
4941          .OVER: MOV          ICNT,@DISPLAY          ;SET UP DISPLAY
4942          TST          LAD          ;FIRST ONE?
4943          BEQ          .SVLAD          ;YES
4944          MOV          LAD,(6)          ;FUDGE RETURN ADDRESS
4945          RTI          ;FIXES PS
4946
4947          TIMES: 1          ;RUN 1 TIMES

```



```

4948 .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
4949
4950 ; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
4951 ; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
4952 ; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
4953 ; "HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
4954 ; (HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADDITIONAL TYPEOUTS.
4955
4956 025406 104464 .HLT:  KBDIN          ; GO CHECK FOR 1G
4957 025410 032777 002000 153410 BIT      #SW10,@SWR      ; BELL ON ERROR?
4958 025416 001402 BEQ      1$          ; NO - SKIP
4959 025420 104402 000007 TYPE     BELL        ; RING BELL
4960 025424 005237 001002 1$:  INC     ERRORS      ; COUNT THE NUMBER OF ERRORS
4961 025430 032777 020000 153370 BIT      #SW13,@SWR      ; SKIP TYPEOUT IF SET
4962 025436 001025 BNE     2$          ; SKIP TYPEOUTS
4963 025440 104402 025444 TYPE     .+2        ; .ASCIZ (<15><12>)
4964 025450 011637 001012 MOV     (6),HLTADR    ; PUT ADDRESS OF INSTRUCTION ON STACK
4965 025454 162737 000002 001012 SUB     #2,HLTADR     ; FUDGE ADDRESS
4966 025462 117737 153324 025544 MOVB   @HLTADR,.HLTCT ; GET HLT ARGUMENT
4967 025470 013746 001012 MOV     HLTADR,-(6)   ; PUT HLTADR ON STACK
4968 025474 104404 TYPE0   TYPE     ; TYPE STACK IN OCTAL
4969 025476 104402 025502 TYPE     .+2        ; .ASCIZ " "
4970 025506 004737 027214 JSR     PC,RSREG     ; GO TO USER ERROR ROUTINE
4971 025512 005777 153310 2$:  TST     @SWR        ; HALT ON ERROR
4972 025516 100001 BPL     .+4         ; SKIP IF CONTINUE
4973 025520 000000 HALT    .+4         ; HALT ON ERROR!
4974 025522 032777 001000 153276 BIT      #SW9,@SWR     ; CHECK FOR INHIBIT LOOP ON ERROR
4975 025530 001003 BNE     3$          ; SKIP IF LOOP ON ERROR
4976 025532 105037 001001 CLRB   ICNT+1       ; CLEAR ITERATION COUNT
4977 025536 000002 RTI     ; RETURN
4978 025540 000137 025356 3$:  JMP     .KIT        ; LOOP ON TEST UNTIL NO ERRORS
4979
4980 025544 000000 .HLTCT: 0          ; HLT ARGUMENT

```

E10

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC
SOCTAL - OCTAL TYPEOUT ROUTINE

MACY11 27(732) 25-SEP-76 10:44 PAGE 122

```

4981          .SBTTL          SOCTAL - OCTAL TYPEOUT ROUTINE
4982
4983          :THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
4984          :ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
4985          :16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.
4986
4987 025546 012737 170101 025734 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
49 9 025554 000411              BR          .PTIT          ;NOW TYPE IT IN BIT FORM
4989 025556 112737 000001 025734 .TYPEO: MOVB     #1 .PR          ;SET ZERO FILL SWITCH
4990 025564 000402              BR          .+6           ;SKIP
4991 025566 005037 025734 .TYPES: CLR      .PR          ;SUPPRESS LEADING ZERO'S
4992 025572 112737 177772 025735 .TYPEO: MOVB     #-6,.PR+1    ;SET COUNT
4993 025600
4994 025600 010446              MOV      R4,-(6)      ;PUSH R4 ON STACK
4995 025602 010546              MOV      R5,-(6)      ;PUSH R5 ON STACK
4996 025604 016605 000010      MOV      10(6),R5     ;GET THE DATA
4997 025610 012704 025736      MOV      #.PR+2,R4   ;SET POINTER TO FIRST ASCII CHAR.
4998 025614 105014              CLRB     (4)         ;CLEAR FIRST BYTE
4999 025616 000411              BR          .PRF      ;ROTATE FIRST BIT
5000 025620 105014              CLRB     (4)         ;CLEAR BYTE OF CHARACTER
5001 025622 032737 000100 025734 .PRL:  BIT      #100,.PR    ;BIT TYPING MODE?
5002 025630 001004              BNE     .PRF         ;YES - SKIP 2 ROTATES
5003 025632 006105              ROL     R5           ;ROTATE BIT INTO C
5004 025634 106114              ROLB    (4)         ;PACK IT
5005 025636 006105              ROL     R5           ;ROTATE BIT INTO C
5006 025640 106114              ROLB    (4)         ;PACK IT
5007 025642 006105              .PRF:  ROL     R5           ;ROTATE BIT INTO C
5008 025644 106114              ROLB    (4)         ;PACK IT
5009 025646 105714              TSTB   (4)         ;IS IT ZERO?
5010 025650 001402              BEQ     .+6         ;SKIP INC
5011 025652 105237 025734      INCB    .PR          ;SET FILL SWITCH
5012 025656 105737 025734      TSTB   .PR          ;CHECK FILL SWITCH
5013 025662 001402              BEQ     .+6         ;SKIP BITSET
5014 025664 152724 000060      BISB   #'0,(4)+     ;MAKE INTO ASCII CHAR
5015 025670 105237 025735      INCB    .PR+1       ;INC COUNT
5016 025674 001351              BNE     .PRL        ;REPEAT
5017 025676 022704 025736      CMP     #.PR+2,R4   ;EMPTY BUFFER?
5018 025702 001002              BNE     .+6         ;SKIP IF NOT
5019 025704 112724 000060      MOVB   #'0,(4)+     ;LOAD 1 ZERO
5020 025710 105014              CLRB   (4)         ;NULL TERMINATOR
5021 025712 104402 025736      TYPE   .PR+2       ;TYPE IT
5022 025716 012605              MOV     (6)+,R5     ;POP STACK INTO R5
5023 025720 012604              MOV     (6)+,R4     ;POP STACK INTO R4
5024 025722 016666 000002 000004 .TYPEO: MOV     2(6),4(6) ;GET RID OF
5025 025730 012616              MOV     (6)+,(6)   ;DATA WORD
5026 025732 000002              RTI
5027
5028 025734 000012          .PR:  .BLKW 12      ;COUNT, SWITCH, AND OUTPUT BUFFER

```

F10

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-REG3 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 123
\$POWER - POWER DOWN AND UP ROUTINES

```

5029          .SBTTL          $POWER - POWER DOWN AND UP ROUTINES
5030
5031          :THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
5032          :THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
5033          :WAIT FOR POWER TO GO DOWN AND BE RESTORED.
5034          :IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS.
5035          :THE PROGRAM WILL HALT AT '.ILLUP'.
5036
5037 025760 012777 026106 000126 .POWER: MOV      #.ILLUP,2.PUVEC ;SET FOR FAST UP
5038 025766 012777 000340 000122      MOV      #340,2.PUVECS+2 ;PRIO:7
5039 025774 010046          MOV      R0,-(6) ;PUSH R0 ON STACK
5040 025776 010146          MOV      R1,-(6) ;PUSH R1 ON STACK
5041 026000 010246          MOV      R2,-(6) ;PUSH R2 ON STACK
5042 026002 010346          MOV      R3,-(6) ;PUSH R3 ON STACK
5043 026004 010446          MOV      R4,-(6) ;PUSH R4 ON STACK
5044 026006 010546          MOV      R5,-(6) ;PUSH R5 ON STACK
5045 026010 010637 026112      MOV      SP,.SAVR6 ;SAVE SP
5046 026014 012777 026024 000072      MOV      #.POWUP,2.PUVEC ;SET UP VECTOR
5047 026022 000000          HALT      ;WAIT FOR PF
5048
5049 026024 013706 026112      .PCWUP: MOV      .SAVR6,SP ;GET SP
5050 026030 005001          CLR      R1 ;WAIT LOOP FOR THE TTY
5051 026032 005201          15: INC     R1 ;WAIT FOR THE INC
5052 026034 001376          BNE     15 ;OF WORD
5053 026036 012605          MOV      (6)+,R5 ;POP STACK INTO R5
5054 026040 012604          MOV      (6)+,R4 ;POP STACK INTO R4
5055 026042 012603          MOV      (6)+,R3 ;POP STACK INTO R3
5056 026044 012602          MOV      (6)+,R2 ;POP STACK INTO R2
5057 026046 012601          MOV      (6)+,R1 ;POP STACK INTO R1
5058 026050 012600          MOV      (6)+,R0 ;POP STACK INTO R0
5059 026052 012737 025760 000024      MOV      #.POWER,2#24 ;SET UP THE POWER DOWN VECTOR
5060 026060 012737 000340 000026      MOV      #340,2#26 ;PRIO:7
5061 026066 104402 026072          TYPE   .,+2 ;.ASCIZ <15><12>"POWER"
5062 026102 000137 021754          JMP     MULSYS ;JMP TO USER ADDRESS
5063
5064 026106 000000          .ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
5065 026110 000776          BR     .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
5066
5067 026112 000000          .SAVR6: 0 ;PUT THE SP HERE
5068 026114 000024 000026      .PUVEC: 24,26 ;POWER UP VECTOR

```



```

S109
S110
S111
S112
S113
S114
S115
S116 026252 010546
S117 026254 012705 026374
S118 026260 022705 026414
S119 026264 001423
S120 026266 105737 177560
S121 026272 100375
S122 026274 113715 177562
S123 026300 142715 000200
S124 026304 122715 000025
S125 026310 001006
S126 026312 104402 026316
S127 026324 000753
S128 026326 122715 000177
S129 026332 001005
S130 026334
S131 026334 104402 026340
S132 026344 000743
S133 026346 111527 000000
S134 026352 104402 026350
S135 026356 122725 000015
S136 026362 001336
S137 026364 104402 000012
S138 026370 012605
S139 026372 000002
S140
S141 026374 000020

```

```

.SBTTL          SRDLIN - TTY INPUT ROUTINE

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

.RDLIN: MOV      R5, -(6)          ;SAVE R5
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

```

```

5142          .SBTTL          STRAP - TRAP HANDLER
5143
5144          ; THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
5145          ; SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
5146          ; THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
5147          ; FOLLOW THIS MACRO.
5148
5149 026414 011646          .TRAP:  MOV      (6) ,-(6)          ; GET ADDRESS OF TRAP +2
5150 026416 162716 000002          SUB      #2,(6)          ; MAKE IT ADDRESS OF TRAP
5151 026422 017616 000000          MOV      2(6) ,(6)          ; GET TRAP INSTRUCTION
5152 026426 062716 122034          ADD      #.TRP+2-TRAP,(6) ; GET DATA AND MAKE IT AN OFFSET
5153 026432 013607          .TRP:  MOV      2(6)+,PC          ; GO TO PROPER SUBROUTINE
5154
5155 026434 025212          .SCOPE          ;SCOPE = TRAP+0 (104400)
5156 026436 025054          .TYPE          ;TYPE = TRAP+2 (104402)
5157 026440 025556          .TYPE0         ;TYPE0 = TRAP+4 (104404)
5158 026442 025566          .TYPES         ;TYPES = TRAP+6 (104406)
5159 026444 026120          .RDOCT         ;RDOCT = TRAP+10 (104410)
5160 026446 026252          .RDLIN         ;RDLIN = TRAP+12 (104412)
5161 026450 027016          .CLRDK         ;CLRDK = TRAP+14 (104414)
5162 026452 027044          .MRDMD         ;MRDMD = TRAP+16 (104416)
5163 026454 022276          .MRCK          ;MRCK = TRAP+20 (104420)
5164 026456 022246          .MRCLK         ;MRCLK = TRAP+22 (104422)
5165 026460 022240          .MRINT         ;MRINT = TRAP+24 (104424)
5166 026462 022342          .DSCK          ;DSCK = TRAP+26 (104426)
5167 026464 022324          .MRIND         ;MRIND = TRAP+30 (104430)
5168 026466 022370          .XBIT          ;XBIT = TRAP+32 (104432)
5169 026470 022660          .CLKD1         ;CLKD1 = TRAP+34 (104434)
5170 026472 022734          .CLKD0         ;CLKD0 = TRAP+36 (104436)
5171 026474 023664          .CLKR1         ;CLKR1 = TRAP+40 (104440)
5172 026476 024046          .CLKR0         ;CLKR0 = TRAP+42 (104442)
5173 026500 023510          .RBIT          ;RBIT = TRAP+44 (104444)
5174 026502 023456          .MCLK1         ;MCLK1 = TRAP+46 (104446)
5175 026504 023500          .MCLK0         ;MCLK0 = TRAP+50 (104450)
5176 026506 023256          .MCLKB         ;MCLKB = TRAP+52 (104452)
5177 026510 027136          .GETSP         ;GETSP = TRAP+54 (104454)
5178 026512 027176          .SPASS         ;SPASS = TRAP+56 (104456)
5179 026514 026524          .SUSWR         ;SUSWR = TRAP+60 (104460)
5180 026516 026726          .CNTLU         ;CNTLU = TRAP+62 (104462)
5181 026520 026644          .KBDIN         ;KBDIN = TRAP+64 (104464)
5182 026522 026726          .CNTLU         ;CNTLU = TRAP+66 (104466)
5183
5184
5185 026524 032737 000001 026642 .SUSWR: BIT      #BIT0,SWI
5186 026532 001037          BNE      XXX
5187 026534 013746 000006          MOV      6,-(SP)          ; SAVE 6 ON STACK
5188 026540 013746 000004          MOV      4,-(SP)          ; SAVE 4 ON STACK
5189 026544 012737 026564 000004          MOV      #1$,4          ; SET UP TRAP ADDRESS
5190 026552 022777 177777 152246          CMP      #-1,2$SWR          ; TEST 177570
5191 026560 001402          BEQ      2$              ; FAKE OUT
5192 026562 000407          BR       3$              ; HARDWARE AVAILABLE
5193 026564 022626          1$:     CMP      (SP)+,(SP)+          ; ADJUST STACK
5194 026566 012737 000176 001026          2$:     MOV      #SWREG,SWR          ; SET UP SOFTWARE REGISTERS
5195 026574 012737 000174 001030          MOV      #DISPRG,DISPLAY
5196 026602 022737 000176 001026          3$:     CMP      #SWREG,SWR          ; 1ST TIME THRU?
5197 026610 001004          BNE      4$              ; NO CHANGE STILL 177570

```


K10

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 25-SEP-76 10:44 PAGE 128
STRAP - TRAP HANDLER

```

5236                                     ;CLEAR ALL DISK REGISTERS
5237 027016 012777 000040 152056 .CLRDK: MOV #40, @RSCS2 ;CLEAR ALL DSK REG
5238 027024 013777 001162 152050          MOV UNNUM, @RSCS2 ;GET UNIT NUMBER
5239 027032 005037 001176          CLR MCCNT ;CLEAR MAINT CLOCK COUNT
5240 027036 005037 001200          CLR MCCNT+2
5241 027042 000002          RTI
5242
5243 027044 012777 000001 152052 .MRDMD: MOV #1, @RSMR ;PUT DRIVE INTO MAINT MODE
5244 027052 000002          RTI
5245
5246 027054 005037 001216 WAITRY: CLR WORK ;CLEAR COUNTER
5247 027060 105777 152014 1$: TSTB @RSCS1 ;TEST READY
5248 027064 100406          BMI 2$ ;OK CONT
5249 027066 005237 001216          INC WORK ;UPDATE COUNTER
5250 027072 005737 001216          TST WORK ;DONE YET?
5251 027076 001403          BEQ 3$ ;READY DID NOT COME UP
5252 027100 000767          BR 1$ ;CONTINUE WAITING
5253 027102 062716 000002 2$: ADD #2, (SP) ;UPDATE RETURN PC
5254 027106 000207          3$: RTS PC ;RETURN
5255
5256                                     ;ROUTINE TO SHIFT COMPLETE DATA TABLE ONE BIT
5257                                     ;TO THE LEFT. CARRIES BIT 15 IF ONE RVD TO BIT 0 OF THE NEXT WORD
5258
5259 027110 012702 030114 MDATA: MOV #INBUF, R2 ;GET LEFT ADDRESS OF
5260 027114 062702 000442          ADD #442, R2 ;DATA TABLE
5261 027120 012703 000220          MOV #220, R3 ;SETUP COUNTER FOR 200 WORDS
5262 027124 000241          CLC ;CLEAR CARRY
5263 027126 006142          1$: ROL -(R2) ;SHIFT DATA PATTERN
5264 027130 005303          DEC R3 ;DO ALL
5265 027132 001375          BNE 1$ ;WORDS
5266 027134 000207          RTS PC
5267 027136 012737 001001 001204 .GETSP: MOV #1001, REPT ;SETUP COUNTER
5268 027144 104430          MRIND ;SEND INDEX PULSE TO MR REG
5269 027146 104422          1$: MRCLK ;CLOCK MR
5270 027150 005337 001204          DEC REPT ;TO REACH
5271 027154 001374          BNE 1$ ;SECTOR PULSE
5272 027156 032777 000400 151740          BIT #400, @RSMR ;DID SECTOR PULSE SET????
5273 027164 001401          BEQ 2$ ;YES
5274 027166 000002          RTI ;NO REPORT ERROR
5275 027170 062716 000002 2$: ADD #2, (SP) ;UPDATE RETURN ADDR
5276 027174 000002          RTI
5277
5278 027176 104422          .SPASS: MRCLK ;CLOCK PAST SECTOR PULSE
5279 027200 104422          MRCLK
5280 027202 005037 001176          CLR MCCNT ;RESET MAINT CLOCK COUNTERS
5281 027206 005037 001200          CLR MCCNT+2
5282 027212 000002          RTI

```



```

5283          :ERROR TYPTXTOUT ROUTINE
5284
5285 027214 005737 025544 RSREG: TST      .HLTCT      ; SHOULD WE TYPTXT GOOD AND BAD
5286 027220 001022          BNE      8$          ; NO
5287 027222 104402 027226          TYPE     .+2          ; .ASCIZ " BAD="
5288 027234 010046          MOV      BAD,-(6)      ; PUT BAD ON STACK
5289 027236 104404          TYPE0    ; TYPE STACK IN OCTAL
5290 027240 104402 027244          TYPE     .+2          ; .ASCIZ " GOOD="
5291 027254 010146          MOV      GOOD,-(6)     ; PUT GOOD ON STACK
5292 027256 104404          TYPE0    ; TYPE STACK IN OCTAL
5293 027260 000402          BR       8$          ; TYPEOUT REGISTERS
5294 027262 000137 027720          JMP      PTDONE      ; GET OUT
5295 027266
5296 027266 104402 027272 8$:          TYPE     .+2          ; .ASCIZ " CS1="
5297 027300 017746 151574          MOV      @RSCS1,-(6) ; PUT @RSCS1 ON STACK
5298 027304 104404          TYPE0    ; TYPE STACK IN OCTAL
5299 027306
5300 027306 104402 027312 1$:          TYPE     .+2          ; .ASCIZ " ER="
5301 027320 017746 151570          MOV      @RSER,-(6)  ; PUT @RSER ON STACK
5302 027324 104404          TYPE0    ; TYPE STACK IN OCTAL
5303 027326
5304 027326 104402 027332 2$:          TYPE     .+2          ; .ASCIZ " CS2="
5305 027340 017746 151536          MOV      @RSCS2,-(6) ; PUT @RSCS2 ON STACK
5306 027344 104404          TYPE0    ; TYPE STACK IN OCTAL
5307 027346 032737 000200 025544          BIT      #200,.HLTCT ; TYPTXT SECOND SET ?
5308 027354 001076          BNE      SEEC        ; YES
5309 027356 032737 000100 025544          BIT      #AS,.HLTCT ; TYPTXT ER ?
5310 027364 001410          BEQ      3$          ; NO
5311 027366 104402 027372          TYPE     .+2          ; .ASCIZ " AS="
5312 027400 017746 151512          MOV      @RSAS,-(6) ; PUT @RSAS ON STACK
5313 027404 104404          TYPE0    ; TYPE STACK IN OCTAL
5314 027406 032737 000020 025544 3$:          BIT      #BA,.HLTCT ; TYPTXT BUS ASSRESS
5315 027414 001410          BEQ      4$          ; NO
5316 027416 104402 027422          TYPE     .+2          ; .ASCIZ " BA="
5317 027430 017746 151452          MOV      @RSBA,-(6) ; PUT @RSBA ON STACK
5318 027434 104404          TYPE0    ; TYPE STACK IN OCTAL
5319 027436 032737 000004 025544 4$:          BIT      #DA,.HLTCT ; TYPTXT DA ?
5320 027444 001410          BEQ      5$          ; NO
5321 027446 104402 027452          TYPE     .+2          ; .ASCIZ " DA="
5322 027460 017746 151424          MOV      @RSDA,-(6) ; PUT @RSDA ON STACK
5323 027464 104404          TYPE0    ; TYPE STACK IN OCTAL
5324 027466 032737 000010 025544 5$:          BIT      #WC,.HLTCT ; TYPTXT WC?
5325 027474 001410          BEQ      6$          ; NO
5326 027476 104402 027502          TYPE     .+2          ; .ASCIZ " WC="
5327 027510 017746 151370          MOV      @RSWC,-(6) ; PUT @RSWC ON STACK
5328 027514 104404          TYPE0    ; TYPE STACK IN OCTAL

```

M10

MAINDEC-11-DZRSE-C
DZRSEC.P11

RS11-RS03 MAINTENANCE MODE DIAGNOSTIC
STRAP - TRAP HANDLER

MACY11 27(732) 25-SEP-76 10:44 PAGE 130

5329	027516	032737	000040	025544	6\$:	BIT	#DS,.HLTCT	;DRIVE STATUS
5330	027524	001475				BEQ	PTDONE	;NO
5331	027526	104402	027532			TYPE	.+2	;ASCIZ "DS="
5332	027540	017746	151346			MOV	ARSDS,-(6)	;PUT ARSDS ON STACK
5333	027544	104404				TYPEO		;TYPE STACK IN OCTAL
5334	027546	000137	027720			JMP	PTDONE	;GET OUT
5335	027552	042737	000200	025544	SEEC:	BIC	#200,.HLTCT	;CLEAR COMMON BIT
5336	027560	032737	000240	025544		BIT	#DT,.HLTCT	;TYPTXT DRIVE TYPE?
5337	027566	001410				BEQ	9\$;NO
5338	027570	104402	027574			TYPE	.+2	;ASCIZ "DT="
5339	027602	017746	151320			MOV	ARSDT,-(6)	;PUT ARSDT ON STACK
5340	027606	104404				TYPEO		;TYPE STACK IN OCTAL
5341	027610	032737	000210	025544	9\$:	BIT	#DB,.HLTCT	;TYPTXT DATA BUFFER
5342	027616	001410				BEQ	10\$;NO
5343	027620	104402	027624			TYPE	.+2	;ASCIZ "DB="
5344	027632	017746	151264			MOV	ARSDB,-(6)	;PUT ARSDB ON STACK
5345	027636	104404				TYPEO		;TYPE STACK IN OCTAL
5346	027640	032737	000220	025544	10\$:	BIT	#MR,.HLTCT	;TYPTXT MN?
5347	027646	001410				BEQ	11\$;NO
5348	027650	104402	027654			TYPE	.+2	;ASCIZ "MR="
5349	027662	017746	151236			MOV	ARSMR,-(6)	;PUT ARSMR ON STACK
5350	027666	104404				TYPEO		;TYPE STACK IN OCTAL
5351	027670	032737	000204	025544	11\$:	BIT	#LA,.HLTCT	;TYPTXT LA?
5352	027676	001410				BEQ	PTDONE	;NO
5353	027700	104402	027704			TYPE	.+2	;ASCIZ "LA="
5354	027712	017746	151202			MOV	ARSLA,-(6)	;PUT ARSLA ON STACK
5355	027716	104404				TYPEO		;TYPE STACK IN OCTAL
5356	027720	052737	100000	001170	PTDONE:	BIS	#BIT15,ONCEE	;SET FORND ERROR FLAG
5357	027726	032737	000040	001170		BIT	#BIT5,ONCEE	
5358	027734	001466				BEQ	1\$	
5359	027736	104402	027742			TYPE	.+2	;ASCIZ <15><12>"MAINT CLOCK COUNT "
5360	027770	013737	001176	001230		MOV	MCCNT,WORK4	;GET MAINT CLOCK COUNT
5361	027776	013737	001200	001224		MOV	MCCNT+2,WORK2	;CAL NUMBERS FOR DOUBLE PRECISION
5362	030004	006137	001224			ROL	WORK2	
5363	030010	006137	001230			ROL	WORK4	
5364	030014	000241				CLC		
5365	030016	013746	001230			MOV	WORK4,-(6)	;PUT WORK4 ON STACK
5366	030022	104406				TYPES		;TYPE STACK IN OCTAL - SUPRESS
5367	030024	012737	000005	001232		MOV	#5,WORK5	
5368	030032	005037	001234		2\$:	CLR	WORK6	
5369	030036	006137	001224			ROL	WORK2	
5370	030042	006137	001234			ROL	WORK6	
5371	030046	006137	001224			ROL	WORK2	
5372	030052	006137	001234			ROL	WORK6	
5373	030056	006137	001224			ROL	WORK2	
5374	030062	006137	001234			ROL	WORK6	
5375	030066	013746	001234			MOV	WORK6,-(6)	;PUT WORK6 ON STACK
5376	030072	104406				TYPES		;TYPE STACK IN OCTAL - SUPRESS
5377	030074	005337	001232			DEC	WORK5	
5378	030100	001354				BNE	2\$	
5379	030102	104402	030106			TYPE	.+2	;ASCIZ <15><12>
5380	030112	000207			1\$:	RTS	PC	
5381	030114	000300			INBUF:	.BLKW	300	
5382	030714	000300			OUTBUF:	.BLKW	300	

```

5383          ; THIS ROUTINE IS FOR PROGRAMMERS ONLY !!!!!!!!!!!!!!! THIS ROUTINE IS USED TO "DETERMINE" A
5384          ; SO THAT A 1 CAN BE ROTATED THROUGH THE CRC REGISTER BY ROTATING THE DATA PATTERN
5385 031514 012737 000040 001144 CRCAL: MOV #40,FLAG2
5386 031522 012706 000500          MOV #500,SP
5387 031526 005037 001226          CLR WORK3
5388 031532 012702 030114          MOV #INBUF,R2
5389 031536 012701 000221          MOV #145.,R1
5390 031542 005022          1$: CLR (R2)+ ; CLEAR DATA BUFFER
5391 031544 005301          DEC R1
5392 031546 001375          BNE 1$
5393 031550 012737 000401 001226          MOV #401,WORK3 ; START WITH A NUMBER OF 401
5394 031556 012702 030114          3$: MOV #INBUF,R2
5395 031562 062702 000100          ADD #100,R2
5396 031566 062737 000003 001226          ADD #3,WORK3
5397 031574 013712 001226          MOV WORK3,(R2) ; PUT NUMBER INTO BUFFER
5398 031600 012701 001001          2$: MOV #513.,R1 ; 513=32 WORDS X 16 BITS
5399 031604 005301          6$: DEC R1
5400 031606 001763          BEQ 3$
5401 031610 012700 000040          MOV #10,R0
5402 031614 012702 030114          MOV #INBUF,R2
5403 031620 062702 000102          ADD #102,R2
5404 031624 000241          CLC
5405 031626 006142          5$: ROL -(R2)
5406 031630 005300          DEC R0
5407 031632 001375          BNE 5$
5408 031634 004737 024430          JSR PC,GENCRC
5409 031640 022737 000001 001216          CMP #1,WORK
5410 031646 001013          BNE 4$
5411 031650 104402 031654          TYPE ,+2 ; .ASCIZ <15><12>"CRC= "
5412 031664 013746 001216          MOV WORK,-(6) ; PUT WORK ON STACK
5413 031670 104404          TYPEO ; TYPE STACK IN OCTAL
5414 031672 004737 031736          JSR PC,TABTYP
5415 031676 022737 000002 001216 4$: CMP #2,WORK
5416 031704 001337          BNE 6$
5417 031706 104402 031712          TYPE ,+2 ; .ASCIZ <15><12>'CRC= "
5418 031722 013746 001216          MOV WORK,-(6) ; PUT WORK ON STACK
5419 031726 104404          TYPEO ; TYPE STACK IN OCTAL
5420 031730 004737 031736          JSR PC,TABTYP
5421 031734 000723          BR 6$
5422 031736 012702 030114          TABTYP: MOV #INBUF,R2
5423 031742 012705 000220          MOV #220,R5
5424 031746 012737 000004 001204 2$: MOV #4,REPT
5425 031754          1$:
5426 031754 012246          MOV (R2)+,-(6) ; PUT (R2)+ ON STACK
5427 031756 104404          TYPEO ; TYPE STACK IN OCTAL
5428 031760 104402 000040          TYPE ,40
5429 031764 005305          DEC R5
5430 031766 001410          BEQ 3$
5431 031770 005337 001204          DEC REPT
5432 031774 001367          BNE 1$
5433 031776 104402 032002          TYPE ,+2 ; .ASCIZ <15><12>
5434 032006 000757          BR 2$
5435 032010 000207          3$: RTS PC
5436
5437          000001          .END

```


CRC TYP	023006	3702	4547#											
CRCMD	024372	4742	4771#											
CS1	= 000001	893#	1142	1201	1241	1245	1249	1257	1474	1478	1533	1537	1664	1669
CS2	= 000200	900#	1133	1197	1267	1277	1281	1286	1530	1542	1559	1584	1595	1607
CTN	027014	5081#	5098#	5228	5234#									
DA	= 000004	895#	1153	1369	1373	1377	1398	2407	2458	2613	2629	2875	2881	3078
		3084	3303	3309	3500	3694	3805	3812	3818	3905	3909	3915	3919	4166
		4169	4175	5319										
DAO	= 001000	921#												
DB	= 000210	902#	5341											
DCK	= 100000	922#												
DISPLA	001030	858#	4937#	4941#	5195#									
DISPRE	000174	798#	5195											
DLT	= 100000	915#	1605											
DONE	021672	1106	4363#											
DONEE	002552	1087	1093	1105#										
DRY	= 000200	916#												
DS	= 000040	898#	1137	2064	2130	2134	2145	2149	2248	2252	2255	2355	2359	2362
		2402	2411	2415	2453	2462	2466	2507	2513	2517	2554	2560	2564	2622
		3078	3303	3500	3694	3700	3733	3751	3805	3812	3818	3824	3864	3868
		387	3878	3900	3905	3915	3919	4158	4172	4258	4261	4266	4269	4313
		432#	4351	4355	5329									
DJCK	= 104426	2230	2236	2243	2305	2311	2350	2380	2397	2431	2448	2485	2502	2534
		2549	4222	5166#										
		904#	5336											
DT	= 000240	1048	1051#											
DVNUM	002036	894#	1159	1218	1406	1410	1417	1423	1430	3751				
ER	= 00 002	919#												
ERR	= 040000	848#	4960#	4980										
ERRORS	001002	4587	4622#											
E1910	023440	4589	4621#											
E1915	023447	4583	4620#											
E192	023424	4585	4621#											
E197	023432	4577	4614#											
E2410	023377	4579	4616#											
E2412	023406	4581	4618#											
E2415	023415	4571	4608#											
E242	023355	4573	4610#											
E245	023363	4575	4612#											
E247	023371	4565	4602#											
E3010	023330	4567	4604#											
E3012	023337	4569	4605#											
E3015	023346	4559	4503#											
E302	023314	4561	4600#											
E305	023322	4563	4601#											
E307	023322	852#	4605	4886										
FILCHR	001014	927#	949#	2653#	2882	2904#	3139#	3324#	3333	3364#	3521#	3530	3559#	3933#
FLAG2	001144	4292#	4450	4456#	4464	4466#	4467#	4471	4475	4479#	4482	4496	4498#	4499#
		4646#	4650	4653#	4658	4660#	4679	4706	4785	4798	4806	5385#		
FLAG3	001146	805#	808#	928#	997#	1001	1009#	1086	4384					
FLOTBA	003616	1307#												
FLOTDA	004116	1382#												
FLOTWC	003756	1345#												
GENCRC	024430	2814	3022	3246	4101	4784#	5408							
GETSP	= 104454	2689	2942	3167	3368	3563	3970	5177#						
GOOD	= %000001	842#	1037#	1272#	1307#	1309	1311	1314#	1345#	1347	1349	1352#	1382#	1384

1386	1389*	1485*	1486*	1488	1526*	1527*	1528	1539*	1540	1552*	1553*	1555
1560*	1562	1565*	1567	1579*	1580*	1582	1585*	1586*	1587	1590	1612*	1613
1621*	1622	1623*	1626*	1628	1629*	1634*	1636	1639*	1642*	1644	1648*	1833*
1834	1845*	1846	1910*	1911	1921*	1922	1933*	1934	1945*	1946	1977*	1978
1999*	2000	2010*	2011	2017*	2018	2055*	2056	2068*	2069	2077*	2078	2085*
2086	2165*	2166*	2168	2180*	2181	2187*	2188*	2189	2195*	2196	2391*	2392
2442*	2443	2496*	2497	2544*	2545	2589*	2590	2594*	2599	2600	2605*	2607
2631*	3091*	3097	3103	3105	3107*	3110*	3117	3741*	3742*	3743	3761*	3770*
3771*	3772	3856*	3857	4429*	4431	4441*	4443	4517*	4518*	4521	4535*	4536*
4539	4678*	4681*	4682*	4687*	4691*	4696*	4698	4705*	4708*	4709*	4712*	4716*
4721*	4723	4748*	4753*	4761*	4766*	4771*	4774*	4777*	5291			
814#	1038	1133	1137	1142	1147	1153	1159	1165	1171	1177	1197	1201
1206	1212	1218	1223	1228	1241	1245	1249	1257	1267	1273	1277	1281
1286	1294	1298	1302	1313	1324	1332	1336	1340	1351	1361	1369	1373
1377	1388	1398	1406	1410	1417	1423	1430	1440	1447	1455	1463	1474
1478	1490	1498	1502	1510	1514	1530	1533	1537	1542	1559	1564	1569
1584	1592	1595	1607	1615	1638	1646	1664	1669	1696	1702	1714	1718
1727	1731	1734	1743	1747	1756	1760	1770	1774	1780	1787	1791	1802
1816	1820	1836	1848	1877	1895	1889	1892	1901	1905	1913	1924	1936
1948	1980	2002	2013	2020	2023	2063	2064	2072	2082	2090	2130	2134
2145	2149	2171	2185	2193	2200	2232	2238	2245	2248	2252	2255	2277
2284	2288	2295	2299	2307	2313	2320	2329	2333	2342	2346	2352	2355
2359	2362	2382	2396	2399	2402	2407	2411	2415	2433	2447	2450	2453
2458	2462	2466	2487	2501	2504	2507	2513	2517	2536	2548	2551	2554
2560	2564	2609	2613	2618	2622	2629	2632	2692	2698	2707	2711	2720
2724	2732	2736	2744	2748	2752	2758	2762	2771	2778	2782	2795	2800
2831	2838	2842	2850	2854	2858	2865	2869	2875	2978	2881	2945	2952
2961	2965	2974	2978	2985	2989	2999	3011	3014	3035	3039	3043	3047
3057	3061	3071	3078	3081	3084	3108	3170	3176	3185	3189	3198	3202
3210	3214	3223	3235	3238	3259	3263	3267	3271	3281	3285	3295	3303
3306	3309	3371	3378	3387	3391	3400	3404	3411	3415	3424	3436	3439
3456	3460	3464	3468	3478	3482	3492	3500	3503	3566	3572	3591	3585
3594	3598	3605	3609	3618	3630	3633	3650	3654	3658	3662	3672	3676
3686	3694	3697	3700	3733	3745	3751	3755	3762	3774	3805	3812	3818
3824	3860	3864	3868	3874	3878	3900	3905	3909	3915	3919	3973	3979
3988	3992	4001	4005	4013	4017	4025	4029	4033	4039	4043	4052	4056
4063	4076	4081	4116	4124	4128	4137	4141	4148	4152	4156	4158	4166
4169	4172	4175	4224	4232	4239	4243	4255	4258	4261	4266	4269	4310
4313	4325	4328	4351	4355								
851#	4964*	4965*	4966	4967	4981							
3799	3826#											
847#	995*	4383*	4915	4930	4932	4934*	4935*	4937	4940*	4941	4947	4976*
924#												
2108	2110#											
946#	4810*	4812*	4814*	4818	4821	4829*	4832	4842*	4845	4861	4862*	4863*
2668	2686	2783	2816	2912	3003	3024	3091	3147	3164	3227	3248	3346
3428	3446	3541	3622	3640	3949	3967	4064	4103	4301*	4303	4318	4343
4344*	4788	5259	5381#	5388	5394	5402	5422					
4291	4359#											
1007	5062	5117	5118	5141#								
1656#												
1665	1670#											
3965	4164#											
4159	4176#											
910#												
4912	4956	5181#										

HLT = 104000

HLTRDR 001012
 IADONE 017176
 ICNT 001000
 IE = 000100
 ILFDON 007566
 INBIT 001212
 INBUF 030114

INFTST 021660
 INPUT 026374
 INT 005444
 INTDON 005534
 INTMR 020560
 INTMR1 020630
 IR = 000100
 KBOIN = 104464

SW12	=	010000	785#	4367	4369		
SW13	=	020000	784#	4961			
SW14	=	040000	783#	4917			
SW15	=	100000	782#				
SW8	=	000400	790#	4912	4913		
SW9	=	001000	789#	4974			
TABTYP		031736	5414	5420	5422#		
TBOIA		021362	4314	4332#			
TIMES		025404	1112	1113*	1237*	4932	4947#
TIMSV		001172	938#	1112*	1237		
TKB		001022	855#	5216			
TKS		001020	854#	5214			
TPB		001024	856#	4886*	4891*		
TPS		001016	853#	4887	4892		
TRE	=	040000	908#	1534			
TRMR		022173	2394	2445	2499	4409#	
TRY		002016	1046#	1050			
TRYNX		002352	1059	1066	1086#	4360	
TSTEVB		022704	4510	4513	4517#		
TST1		002606	1117#				
TST10		003646	1316#				
TST11		003672	1325#				
TST12		003754	1341#				
TST13		004006	1355#				
TST14		004032	1362#				
TST15		004114	1378#				
TST16		004146	1392#				
TST17		004172	1399#				
TST2		003030	1181#				
TST20		004234	1412#				
TST21		004256	1418#				
TST22		004300	1424#				
TST23		004324	1431#				
TST24		004362	1441#				
TST25		004410	1448#				
TST26		004446	1456#				
TST27		004504	1467#				
TST3		003252	1232#				
TST30		004556	1480#				
TST31		004622	1492#				
TST32		004672	1504#				
TST33		004744	1519#				
TST34		005046	1546#				
TST35		005154	1570#				
TST36		005326	1616#				
TST37		005442	1655#				
TST4		003344	1250#				
TST40		005534	1674#				
TST41		006270	1852#				
TST42		007044	2027#				
TST43		007566	2114#				
TST44		007756	2154#				
TST45		010330	2209#				
TST46		010512	2259#				
TST47		011050	2366#				
TST5		003372	1262#				

B12

MAINDEC-11-DZSE-3
DZSEC.P11

RS11-RSC3 MAINTENANCE MODE DIAGNOSTIC
CROSS REFERENCE TABLE -- MACRO NAMES

MACY11 27(732) 25-SEP-76 10:44 PAGE 147

STYPED 18
SUMR 18

	1481	1492	1493	1504	1505	1516	1520	1543	1547	1570	1571	1616	1617	1652	1556
	1671	1675	1849	1853	2024	2028	2059	2081	2090	2111	2115	2151	2155	2171	2184
	2192	2199	2206	2210	2256	2260	2363	2367	2396	2416	2420	2447	2468	2472	2501
	2518	2522	2548	2565	2569	2638	2640	2644	2885	2889	3115	3120	3124	3310	3314
	3509	3513	3702	3709	3713	3777	3781	3827	3831	3860	3874	3879	3883	3921	3925
	4177	4181	4282	4286	4334	4351	4369	4382	4395	4556	4964	4970	5062	5108	5127
	5132	5142	5155	5156	5157	5158	5159	5160	5161	5162	5163	5164	5165	5166	5167
	5168	5169	5170	5171	5172	5173	5174	5175	5176	5177	5178	5179	5180	5181	5182
	5183	5221	5223	5226	5288	5291	5297	5301	5305	5312	5317	5322	5327	5332	5339
	5344	5349	5354	5360	5380	5412	5418	5434							
.PAGE	812	845	4905	4948	4981	5029	5069	5109	5142						
.REM	1														
.REPT	792														
.SBTTL	1114	1178	1229	1259	1464	1516	1543	1652	1671	1849	2024	2111	2151	2206	2256
	2363	2416	2468	2518	2565	2640	2885	3120	3310	3509	3709	3777	3827	3879	3921
	4177	4282	4361	4865	4905	4948	4981	5029	5069	5109	5142				
.TITLE	775														

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DZRSEC.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZRSCE.SML,DZRSEC.P11
RUN-TIME: 30 49 7 SECONDS
RUN-TIME RATIO: 125/88=1.4
CORE USED: 23K (45 PAGES)

H12

Speaker pultrine 10 Records 20 MS, 5K1 disk reads, 0 disk writes, 149 pages

0000000111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990000000001111111111222222222233312
567890123456789012 *
0000000111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990000000001111111111222222222233312