

RH70/RS04

MAINT MODE DIAGNOSTIC
MD-11-DERSD-B

EP-DERSD-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN US

The image displays a grid of 100 small diagnostic tables, arranged in 10 rows and 10 columns. Each table contains technical data, likely related to the RH70/RS04 system, such as error codes, status indicators, and diagnostic procedures. The text is small and difficult to read, but the layout is consistent across all cells.

CONTENTS

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

47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

92
93
94
95
96
97
98
99
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

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 R504 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.

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 R504 LOGIC IS REQUIRED TO UTILIZE THIS DIAGNOSTIC EFFECTIVELY. REFER TO SECTIONS 2 AND 3 OF THE "R504 DECDISK SERVICE MANUAL" (DEC-00-HRS4A-A-D) FOR DESCRIPTIONS OF THE DRIVE LOGIC.

2. REQUIREMENTS

2.1 EQUIPMENT

E01

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 5

148
149

PDP-11 WITH A MINIMUM OF 8K OF MEMORY AND AN RH11 CONTROLLER WITH A
RS04 DISK.

150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205

MAINDEC-11-DERSD-B RH11-RS04 BASIC FUNCTION DIAGNOSTIC
DESCRIPTION

PAGE 4

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)

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
RS04 DRIVES ON THE SYSTEM

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

TYPE UNIT #

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

Handwritten scribbles in the top right corner.

GO1

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 7

206
207

"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)
- B. PRESS START
- C. THE PROGRAM WILL THEN TEST ALL RS04 DRIVES ON THE SYSTEM.

5. OPERATIONAL SWITCH SETTINGS

SWITCH SETTINGS ARE:

- SW<15> = 1 HALT ON ERROR
- SW<14> = 1 LOOP ON TEST
- SW<13> = 1 INHIBIT TYPEOUTS
- SW<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 BRIEF DESCRIPTION OF WHAT EACH ONE DOES.

5.1.1 CLDK

TRAPS TO A TAG CALLED ".CLDK". THIS ROUTINE CLEARS ALL REGISTERS BY SETTING THE "CLEAR BIT" IN RSCS2. (MOV#40, RHC2) THE NUMBER OF THE UNIT UNDER TEST IS THEN RELOADED INTO RSCS2 AND THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE CLDK 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

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

I01

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 9

264

NEXT INSTRUCTION FOLLOWING THE MRDMD INSTRUCTION.

MAINDEC-11-DERSD-B RH11-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 6
DESCRIPTION

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.

5.1.5 MRCLK

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 THE TWO DATA BITS THAT ARE CURRENTLY BEING WRITTEN FROM THE DATA BUFFER IN CORE AND STORES ONE BIT IN A LOCATION CALLED NOWOD AND THE OTHER BIT IN LOCATION NOWEV. THE PREVIOUS CONTENTS OF NOWOD AND NOWEV ARE STORED IN LASTOD AND LASTEV, RESPECTIVELY. THIS INFORMATION IS USED BY THE CLKD1 AND CLKD2 ROUTINES TO DETERMINE THE CORRECT STATES OF THE MWDB (BIT 12) AND MWDI (BIT 14) IN BITS IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS04 WRITES 18 BIT WORDS)

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

K01

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-R504 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 11

321
322

EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE
XBIT INSTRUCTION.

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-DERSD-B RH11-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 7
DESCRIPTION

5.1.9 CLKD1 AND CLKD2

TRAPS TO LOCATIONS ".CLKD1" AND ".CLKD2". THESE TWO ROUTINES USE THE DATA BITS RECEIVED FROM THE XBIT ROUTINE TO DETERMINE THE CORRECT STATES OF MWDB (BIT 12) AND MWDT (BIT 14) 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

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

5.1.11 CLKR1 AND CLKR2

TRAPS TO LOCATIONS ".CLKR1" AND ".CLKR2". THESE TWO ROUTINES USING THE DATA BITS RECEIVED FROM THE RBIT ROUTINE SET AND CLEAR THE MRDB (BIT 2) AND MRDT (BIT 5) BITS 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 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 TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.1.13 HLT

MO1

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-R504 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 13

379
380

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

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

MAINDEC-11-DERSD-B RH11-RS04 BASIC FUNCTION DIAGNOSTIC PAGE 8
DESCRIPTION

5.1.14 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

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

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, RSDB, 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, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

4. TEST "CLEAR BIT" IN RSCS2

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

5. LOAD RSDB WITH ALL ONES AND ALL ZEROS

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

6. TEST PROGRAM INTERRUPT

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

7. MAINTENANCE TIMING TEST

THE FOLLOWING TEST ON THE RS04 DISK IS A SINGLE-STEPPED MAINTENANCE MODE TEST ON THE RS04 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,

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

C02

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RSC4 MAINTENANCE MODE DIAGNOSTIC

MACY11 27(732) 04-OCT-76 13:11 PAGE 16

492

ETC.

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

- 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.

8. 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.

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

9. 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.

10. 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.

11. 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

E02

MAINDEC-11-DERSD-8
DERSDB.P11

RS11-R504 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 18

549
550

(DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE
ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.

MAINDEC-11-DERSD-B RH11-R504 BASIC FUNCTION DIAGNOSTIC PAGE 11
DESCRIPTION551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606

12. 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.

13. 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).

14. 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 TRIES TO LOAD ITS REGISTER WITH ALTERNATE ONES AND ZEROS. THIS SHOULD CAUSE "NED" TO SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.

15. MAINTENANCE WRITE TEST

THIS IS AN R504 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.

16. MAINTENANCE READ TEST

THIS IS AN R504 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.)

17. MAINTENANCE MODE DATA WRITE CHECK TEST

A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION. WITHIN THE R504, A WRITE CHECK FUNCTION IS IDENTICAL TO A READ FUNCTION.

G02

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-R504 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 20

607

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

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

THE RS04 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.

19. 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.

20. 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.

21. INVALID ADDRESS TEST

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).

22. 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 (RSDS).

23. 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.

24. MAINTENANCE MODE INTERRUPT TEST

664
665

IN THIS TEST THE INTERRUPT ENABLE (I.E.,) BIT IS SET. A TWO

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

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.

25. 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.

26. 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.

%
:TITLE MAINDEC-11-DEFS0-B RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
:COPYRIGHT 1974,1975,1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
:PROGRAM BY STANLEY HARACKIEWICZ

				SWITCH	USE
				-----	-----
				SW15= 100000	;HALT ON ERROR
				SW14= 40000	;LOOP ON TEST
				SW13= 20000	;INHIBIT ERROR TYPEOUTS
				SW12= 10000	;TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
				SW11= 4000	;RUN MAINTENANCE MODE VERIFY TEST
				SW10= 2000	;0 - BELL ON PASS COMPLETE
					;1 - BELL ON ERROR
				SW9= 1000	;LOOP ON ERROR
				SW8= 400	;LOOP ON TEST IN SW<7:0>
				.= 0	;TRAP CATCHER FROM 0 - 776
				.= 200	
				JMP @#BEGIN1	
				.= 220	
				BIS #BIT6,FLAG2	;TEST ALL DRIVES
				BEGIN2: JMP @#BEGIN	
				BEGIN1: BIC #BIT6,FLAG2	;CLEAR MULTI DRIVE FLAG
				BR BEGIN2	

100000
040000
020000
010000
004000
002000

001000
000400
000000
000200
000200 000137 000232

000220
000220 052767 000100 000716
000226 000137 001234

000232 042767 000100 000704
000240 000772

719
720 000001
721 104000
722 177776
723 177776
724 177570
725 177570
726 000007
727 000000
728 000001
729 000002
730 000003
731 000004
732 000005
733 000006
734 000007
735 000001
736 000002
737 000004
738 000010
739 000020
740 000040
741 000100
742 000200
743 000400
744 001000
745 002000
746 004000
747 010000
748 020000
749 040000
750 100000
751 000001
752 000000
753

N= 1
HLT= EMT
PS= 177776
PSW= PS
SWR= 177570
DISPLAY=SWR
BELL= 7
R0= %0
R1= %1
R2= %2
R3= %3
R4= %4
R5= %5
SP= %6
PC= %7
BIT0= 1
BIT1= 2
BIT2= 4
BIT3= 10
BIT4= 20
BIT5= 40
BIT6= 100
BIT7= 200
BIT8= 400
BIT9= 1000
BIT10= 2000
BIT11= 4000
BIT12= 10000
BIT13= 20000
BIT14= 40000
BIT15= 100000

GOOD= %1
BAD= %0

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

;FOR GOOD DATA
;FOR BAD DATA

754 001000 001000
755
756 001000 000000
757 001002 000000
758 001004 000000 000000
759 001010 000000
760 001012 000000
761 001014 001000
762 001016 177564
763 001020 177566
764
765 001100
766
767
768
769 001100 172040
770 001102 172050
771 001104 172042
772 001106 172044
773 001110 172046
774 001112 172052
775 001114 172054
776 001116 172056
777 001120 172060
778 001122 172062
779 001124 172064
780 001126 172066
781 001130 000204
782 001132 000206
783 001134 172041
784 001136 172051
785 001140 172043
786 001142 172045
787

.= 1000
ICNT: 0
ERRORS: 0
PCNT: 0,0
LAD: 0
HLTADR: 0
FILCHR: 1000
TPS: 177564
TPB: 177566
.
.= 1100
;DISK I/O REGISTERS
RSCS1: 172040
RSCS2: 172050
RSWC: 172042
RSBA: 172044
RSDA: 172046
RSDS: 172052
RSER: 172054
RSAS: 172056
RSLA: 172050
RSDB: 172062
RSMR: 172064
RSDT: 172066
RSVEC: 204
RSVCPS: 206
RSCS1B: 172041
RSCS2B: 172051
RSWCB: 172043
RSBAB: 172045

;LH = ITERATION COUNT ;RH = TEST NO.
;ERROR COUNT
;2 WORD PASS COUNT
;LOOP ADDRESS FOR SCOPE
;ADDRESS OF LAST H.T INSTRUCTION EXECUTED
;FILCHR=0 (CHAR) ;FILCHR+1=2 (COUNT)
;OUTPUT STATUS REGISTER
;OUTPUT BUFFER
;
;DISK CONTROL + STATUS REGISTER
;DISK CONTROL + STATUS REGISTER
;WORD COUNT REGISTER
;BUS ADDRESS
;DISK ADDRESS (DESIRED ADDRESS)
;DRIVE STATUS
;ERROR REG.
;ATTENTION SUMMARY
;LOOK AHEAD
;DATA BUFFER REGISTER
;MAINTENANCE REGISTER
;DRIVE TYPE REGISTER
;INTERUPT VECTOR
;INTERUPT PRIO. VECTOR
;ODD BYTE ADD FOR CS1
;ODD BYTE ADD FOR CS2
;ODD BYTE ADD FOR CW
;ODD BYTE ADD FOR BA

788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829

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
    
```

830
831
832 001144 000000
833 001146 000000
834 001150 000000
835 001152 000000
836 001154 000000
837 001156 000000
838 001160 000000
839 001162 000000
840 001164 000000
841 001166 000000
842 001170 000000
843 001172 000000
844 001174 000000
845 001200 000000
846 001202 000000
847 001204 000000
848 001206 000000
849 001210 000000
850 001212 000000
851 001214 000000
852 001216 000000
853 001220 000000
854 001222 000000
855 001224 000000
856 001226 000000
857 001230 000000
858 001232 000000
859 001232 000000

000000

;WORKING LOCATIONS

FLAG2: 0
LSTEV: 0
LSTOD: 0
NOWEV: 0
NOWOD: 0
RSO: 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

;SECOND FLAG WORD
;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

860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887

;DISCRIPTION OF BITS IN LOCATION ONCEE

;BIT0 MEANS FOUND DRIVE
;BIT1 ERROR DO NOT CHANGE ILLEGAL FUNCTION
;BIT2 ERROR FLAG
;BIT3 TESTING CODE 21 FLAG
;BIT4 TEST ONLY ONE DRIVE
;BIT5 TYPEOUT CLOCK COUNT
;BIT6 1ST TRANSFER WORD FLAG
;BIT7 WRITTING LAST WORD OF SECOTR
;BIT8 TRANSFERRING CRC WORD
;BIT9 FOR INTERLEAVED DRIVES
;BIT10 1ST TIME FLAG IN SECTOR FRACTION TEST
;BIT11 DO TKSEL TEST
;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
;BIT15 MEANS ERROR FOUND

;DISCRIPTION OF BITS IN LOCATION FLAG2

;BIT0 SWITCH FOR RWCLK IN MR REG
;BIT1 MAINTENANCE MODE VERIFY TEST
;BIT2 IN WRITE CK TEST FOR CLKRI ROUTINE
;BIT3 DONE 1ST CRC WD IN CRC TEST
;BIT4 1ST TIME THROUGH IN CRC TEST
;BIT5 IN CRC TEST
;BIT6 FLAG TO TEST ALL DRIVES

```

888 001234 012706 000500          BEGIN:  MOV    #500,SP          ;SET STACK TO *** 500 ***
889 001240 012737 025000 000024    MOV    #.POWER, @#24      ;SET UP PF VECTOR
890 001246 012737 000340 000026    MOV    #340, @#26        ;LOCK OUT THE WORLD
891 001254 012737 024430 000030    MOV    #.HLT, @#30       ;SET EMT VECTOR
892 001262 012737 000340 000032    MOV    #340, @#32       ;LOCK UP
893 001270 012737 025402 000034    MOV    #.TRAP, @#34     ;SET TRAP VECTOR
894 001276 012737 000340 000036    MOV    #340, @#36       ;LOCK UP
895 001304 005067 177470          CLR    ICNT              ;INIT ICNT
896 001310 005067 177474          CLR    LAD              ;INIT LAD
897 001314 042767 177677 177622    BIC    #177677, FLAG2
898 001322 042767 153777 177636    BIC    #153777, ONCEE
899 001330 032767 000100 177606    BIT    #BIT6, FLAG2     ;TEST ALL DRIVES?
900 001336 001402          BEQ    $S              ;ASK
901 001340 000137 001672          JMP
902 001344          SS:
903 001344 104402 001350          TYPE    ,.+2           ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
904 001406 104412          RDLIN
905 001410 122767 000131 023744    CMPB   #'Y, INPUT       ;TEST FOR YES
906 001416 001525          BEQ    MULTII          ;YES
907 001420 052767 000020 177540    BIS    #BIT4, ONCEE     ;SET TEST ONLY ONE DRIVE FLAG
908 001426          IS:
909 001426 104402 001432          TYPE    ,.+2           ;.ASCIZ "TYPE UNIT #"
910 001446 104410          RDOCT
911 001450 012604          MOV    (6)+, R4         ;GET NUMBER
912 001452 022704 000010          CMP    #10, R4         ;CORRECT #
913 001456 101763          BLOS   $S              ;NO
914 001460 010467 177474          MOV    R4, UNNUM       ;SET UNIT #
915 001464 005002          CLR    R2              ;CLEAR WORK AREA
916 001466 000261          SEC
917 001470 006102          2S:  ROL    R2          ;SET CARRY
918 001472 005704          TST   R4              ;SET WORK BIT
919 001474 001402          BEQ    $S              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
920 001476 005304          DEC   R4              ;YES
921 001500 000773          BR    2S              ;NO TRY AGAIN
922 001502 010267 177454          3S:  MOV    R2, UNITSV    ;TEST AGAIN
923 001506 010267 177452          MOV    R2, UNCOMP      ;SET DRIVE BIT IN UNITSV
924 001512 016777 177442 177362    MOV    UNNUM, @RSCS2   ;SET UNIT COMPARE
925 001520 012777 177777 177366    MOV    #-1, @RSER     ;LOAD DRIVE
926 001526 104402 001532          MOV    TYPE, @RSER    ;LOAD ERRORS
927 001642 000000          HALT
928 001644 026777 177312 177244    CMP    UNITSV, @RSAS   ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
929 001652 001405          BEQ    $S              ;WAIT FOR LIGHTS TO BE CHECKED
930 001654 017700 177236          MOV    @RSAS, BAD     ;DID CORRECT ATA SET
931 001660 016701 177276          MOV    UNITSV, GOOD   ;GET RSAS
932 001664 104000          HLT
933
934
935 001666 000167 000430          4S:  JMP    NOWGO          ;GET CORRECT AND
;RSAS=BAD GOOD=CORRECTIONS
;ATA BIT SHOULD SET FOR ERRORS
;WERE SET IN RSER
;START TESTING

```



```

936                                     ;NOW TEST FOR DRIVES
937
938 001672 012701 000010          MULTII: MOV      #8, R1          ;PUT 8 INTO R1 FOR COUNT
939 001676 005077 177200          CLR      @RSCS2        ;SET DEVICE TO ZERO
940 001702 012777 177777 177204  TRY:  MOV      #-1, @RSER       ;CAUSE AN ERROR +SETS BIT IN RSAS REG
941 001710 005301                   DEC      R1            ;DO A MAXIMUM OF 8 TIMES
942 001712 001403                   BEQ     DVNUM          ;TESTED FOR ALL DRIVES GET OUT
943 001714 005277 177162          INC     @RSCS2        ;INCREMENT DRIVE UNIT
944 001720 000770                   BR      TRY           ;REPEAT FOR NEXT DRIVE
945 001722 017767 177170 177232  DVNUM: MOV     @RSAS, UNITSV ;SAVE
946 001730 012767 000401 177226  MOV     #401, UNCOMP  ;SETUP TO CMP WITH UNITSV
947 001736 012767 000000 177214  MOV     #0, UNNUM     ;PUT 0 INTO UNIT NO.
948 001744 032767 020000 175616  BIT     #BIT13, SWR   ;INHIBIT TYPE OUT?
949 001752 001015                   BNE    STTEST        ;YES
950 001754 104402 001760                   TYPE   ,.+2         ;.ASCIZ <15><12>"TESTING UNIT "
951 002000 042767 100000 177160  BIC     #BIT15, ONCEE ;CLEAR ERROR FLAG
952 002006 036767 177152 177146  STTEST: BIT    UNCOMP, UNITSV ;IS THIS DRIVE ON THE SYSTEM
953 002014 001440                   BEQ    TRYNX         ;NO
954 002016 016777 177136 177056  MOV     UNNUM, @RSCS2 ;YES PUT UNIT # INTO CS2
955 002024 022777 000002 177074  3$:    CMP     #2, @RSDT ;IS THIS A RS04?
956 002032 001404                   BEQ    1$           ;YES
957 002034 022777 000003 177064  CMP     #3, @RSDT   ;IS IT A RS04?
958 002042 001025                   BNE    TRYNX        ;GET A NEW NUMBER
959 002044 032767 020000 175516  1$:    BIT     #BIT13, SWR ;INHIBIT TYPE OUT?
960 002052 001020                   BNE    4$           ;YES
961 002054 032767 100000 177104  BIT     #BIT15, ONCEE ;ANY ERRORS?
962 002062 001404                   BEQ    5$           ;NO
963 002064 104402 002070                   TYPE   ,.+2         ;.ASCIZ <15><12><12>
964 002074
965 002074 016746 177060          5$:    MOV     UNNUM, -(6) ;PUT UNNUM ON STACK
966 002100 104406                   TYPES  ;TYPE STACK IN OCTAL - SUPRESS
967 002102 104402 000040          TYPE   ,40         ;TYPE SPACE
968 002106 042767 100000 177052  BIC     #BIT15, ONCEE ;CLEAR ERROR FLAG
969 002114 000502                   BR     NOWGO        ;NOW TEST
970 002116 032767 000020 177042  TRYNX: BIT    #BIT4, ONCEE ;MULTI DRIVE
971 002124 001074                   BNE    DONEE        ;NO
972 002126 006367 177032          1$:    ASL     UNCOMP    ;CHECK NEXT BIT FOR DRIVE
973 002132 103403                   BCS    CHCKDV       ;DID WE TEST ANY REG?
974 002134 005267 177020          INC     UNNUM       ;INC UNIT #
975 002140 000722                   BR     STTEST       ;CHECK FOR NEXT DRIVE

```

E03

MAINDEC-11-DERSO-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC

MACY11 27(732) 04-OCT-76 13:11 PAGE 31

```

976 002142 032767 000001 177016 CHCKDV: BIT      #BIT0,ONCEE      ;DID WE TEST ANY DRIVES?
977 002150 001062          BNE      DONEE        ;YES WE DID TEST A DRIVE
978 002152 012767 100000 177004      MOV      #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
979 002160 005067 176774          CLR      UNNUM         ;ANY AS BITS, THUS DEFAULTS TO
980 002164 032767 020000 175376      BIT      #BIT13,SWR    ;INHIBIT TYPE OUT?
981 002172 001050          BNE      4$           ;YES
982 002174 016746 176760      MOV      UNNUM,-(6)   ;PUT UNNUM ON STACK
983 002200 104406          TYPES          ;TYPE STACK IN OCTAL - SUPRESS
984 002202 104402 000040      TYPE      ,40        ;TYPE SPACE
985 002206 104402 002212      TYPE      ,+2        ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
986 002304 012767 000001 176652      MOV      #1,UNCMP
987 002312 000000          HALT
988 002314 000402          4$: BR      NOWGO
989 002316 000167 016612      DONEE: JMP      DONE     ;GET OUT
990
991          ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
992          ;TO CLEAR ALL THE RH AND RS REGISTERS
993
994 002322 052767 000001 176636 NOWGO: BIS      #BIT0,ONCEE ;SET FOUND DRIVE FLAG
995 002330 016767 022072 176632      MOV      TIMES,TIMSV  ;SAVE TIME
996 002336 012767 000001 022062      MOV      #1,TIMES    ;ONLY TEST ONCE
997
998          ;*****
999          ;TEST 1      RESET TEST FOR REGISTERS
1000          ;*****
1001 002344 104400      TST1: SCOPE
1002 002346 012737 000340 177776      MOV      #340,APS     ;LOCK OUT INTERUPTS
1003 002354 016777 176600 176520      MOV      UNNUM,ARSCS2 ;LOAD UNIT #
1004 002362 012777 177776 176510      MOV      #177776,ARSCS1 ;SET ALL
1005 002370 012777 177777 176510      MOV      #177777,ARSCS1 ;POSSIBLE R/W
1006 002376 012777 177777 176504      MOV      #177777,ARSDA ;BITS IN THESE REGISTERS
1007 002404 012777 177777 176502      MOV      #177777,ARSER
1008 002412 012777 177777 176504      MOV      #177777,ARSMR
1009 002420 012777 177777 176456      MOV      #177777,ARSMC
1010 002426 012777 177737 176446      MOV      #177737,ARSCS2
1011 002434 000005          RESET          ;CLEAR ALL BITS IN ALL REG.
1012
1013          ;TEST RSCS2 FOR CLEARED BITS
1014 002436 022777 000100 176436      CMP      #100,ARSCS2  ;DID THESE BITS GET CLEARED?
1015 002444 001401          BEQ      +4          ;YES
1016 002446 104200          HLT      !CS2        ;(417) SHOULD BE CLEARED IN CS2
1017 002450 016777 176504 176424      MOV      UNNUM,ARSCS2 ;PUT # OF UNIT IN TEST IN CS2
1018 002456 022777 010600 176426      CMP      #10600,ARSDS ;IS DPR AND MOL SET?
1019 002464 001401          BEQ      +4          ;YES
1020 002466 104040          HLT      !DS         ;NO WHY NOT?
1021
1022          ;TEST CONTROL AND STATUS REG 1
1023 002470 022777 004200 176402      CMP      #4200,ARSCS1 ;DID THE READY BIT SET?
1024 002476 001401          BEQ      +4          ;YES
1025 002500 104001          HLT      !CS1        ;READY SHOULD BE SET

```



```

1026 ;TEST BUS ADDRESS REGISTER
1027
1028 002502 005777 176400 TST @RSBA ;IS BA REG. CLEARED
1029 002506 001401 BEQ .+4 ;YES
1030 002510 104020 HLT !BA ;SHOULD BE 0
1031
1032 ;TEST DISK ADDRESS REGISTER
1033
1034 002512 005777 176372 TST @RSDA ;IS DA CLEARED
1035 002516 001401 BEQ .+4 ;YES
1036 002520 104004 HLT !DA ;SHOULD BE 0
1037
1038 ;TEST ERROR REG RSER
1039
1040 002522 005777 176366 TST @RSER ;DID RSER CLEAR?
1041 002526 001401 BEQ .+4 ;YES
1042 002530 104002 HLT !ER ;BITS(157015) SHOULD BE CLEARED
1043
1044 ;TEST RS MAINTENANCE REGISTER
1045
1046 002532 032777 000077 176364 BIT #77,@RSMR ;DID THESE BITS GET CLEARED
1047 002540 001401 BEQ .+4 ;YES
1048 002542 104220 HLT !MR ;BITS(77) SHOULD BE 0
1049
1050 ;TEST WC REG IT SHOULD NOT CHANGE
1051
1052 002544 022777 177777 176332 CMP #177777,@RSWC ;DID IT CHANGE?
1053 002552 001401 BEQ .+4 ;NO
1054 002554 104010 HLT !WC ;RESET SHOULD NOT MODIFY RSWC
1055
1056 ;TEST RSAS
1057
1058 002556 005777 176334 TST @RSAS ;IS REG CLEAR
1059 002562 001401 BEQ .+4 ;YES
1060 002564 104100 HLT !AS ;NO

```

```

1061 ;*****
1062 ;TEST 2 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
1063 ;*****
1064 002566 104400 TST2: SCOPE
1065
1066 002570 012737 000340 177776 TTAGG: MOV #340, @#PS ;LOCK OUT INTERRUPTS
1067 002576 016777 176356 176276 MOV UNNUM, @RSCS2
1068 002604 012777 043576 176266 MOV #43576, @RSCS1 ;SET ALL
1069 002612 012777 177777 176266 MOV #177777, @RSBA ;POSSIBLE
1070 002620 012777 177777 176262 MOV #177777, @RSDA ;REGISTERS
1071 002626 012777 177017 176260 MOV #177017, @RSER
1072 002634 012777 177777 176260 MOV #177777, @RSDB
1073 002642 012777 177777 176234 MOV #177777, @RSWC
1074 002650 012777 020417 176224 MOV #20417, @RSCS2
1075 002656 012777 000071 176240 MOV #71, @RSMR
1076 002664 012777 000040 176210 MOV #40, @RSCS2 ;CLEAR ALL BITS
1077 002672 022777 000100 176202 CMP #100, @RSCS2 ;DID THE RIGHT BITS CLEAR?
1078 002700 001401 BEQ +4 ;YES
1079 002702 104200 HLT !CS2 ;(417) SHOULD BE CLEARED IN CS2
1080 002704 016777 176250 176170 MOV UNNUM, @RSCS2 ;GET DRIVE NUMBER
1081 002712 032777 173577 176160 BIT #173577, @RSCS1 ;DID ALL BITS GET CLEARED
1082 002720 001401 BEQ +4 ;YES
1083 002722 104001 HLT !CS1 ;NO, ALL BITS SHOULD BE 0
1084
1085 ;TEST BUS ADDRESS REGISTER
1086 002724 005777 176156 TST @RSBA ;IS BA REG. CLEARED
1087 002730 001401 BEQ +4 ;YES
1088 002732 104020 HLT !BA ;SHOULD BE 0
1089
1090 ;TEST DISK ADDRESS REGISTER
1091
1092 002734 005777 176150 TST @RSDA ;IS DA CLEARED
1093 002740 001401 BEQ +4 ;YES
1094 002742 104020 HLT !BA ;SHOULD BE 0
1095
1096 ;TEST ERROR REG RSER
1097
1098 002744 032777 177777 176142 BIT #177777, @RSER ;DID THESE BITS GET CLEARED
1099 002752 001401 BEQ +4 ;YES
1100 002754 104002 HLT !ER ;BITS(157015) SHOULD BE CLEARED
1101
1102 ;TEST RS MAINTENANCE REGISTER
1103 002756 032777 000077 176140 BIT #77, @RSMR ;DID THESE BITS GET CLEARED
1104 002764 001401 BEQ +4 ;YES
1105 002766 104220 HLT !MR ;BITS(77) SHOULD BE 0
1106
1107 ;TEST WC REG. IT SHOULD NOT CHANGE
1108 002770 022777 177777 176106 CMP #177777, @RSWC ;DID WC CHANGE
1109 002776 001401 BEQ +4 ;NO
1110 003000 104010 HLT !WC ;WHY DID IT CHANGE?

```


H03

MAINDEC-11-DERSD-B
DERSDB.P11 TST3

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
SET AND CLEAR ALL REGISTERS

MACY11 27(732) 04-OCT-76 13:11 PAGE 34

```
1111 ;*****
1112 ;TEST 3      SET AND CLEAR ALL REGISTERS
1113 ;*****
1114 003002 104400 TST3: SCOPE
1115 ;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
1116 ;BITS 7,6,5,4,3,2&1
1117
1118 003004 104414 CLRDK ;CLEAR ALL RS REG
1119 003006 016767 176156 021412 MOV TIMSV,TIMES ;GET TIME
1120 003014 012777 003576 176056 MOV #3576,RSCS1 ;SET DISK FUNCTION BITS
1121 003022 022777 005776 176050 CMP #5776,RSCS1 ;ARE THESE BITS SET?
1122 003030 001401 BEQ +4 ;NO
1123 003032 104001 HLT !CS1 ;SHOULD = 3776
1124 003034 012777 002524 176036 MOV #2524,RSCS1 ;SET THESE BITS
1125 003042 022777 004724 176030 CMP #4724,RSCS1 ;DID THEY SET
1126 003050 001401 BEQ +4 ;YES
1127 003052 104001 HLT !CS1 ;SHOULD BE 2725
1128 003054 012777 001052 176016 MOV #1052,RSCS1 ;SET THESE BITS
1129 003062 022777 005252 176010 CMP #5252,RSCS1 ;ARE THEY =?
1130 003070 001401 BEQ +4 ;YES
1131 003072 104001 HLT !CS1 ;SHOULD = 1252
1132 003074 104400 TST4: SCOPE
1133 ;CLEAR THE FUNCTION BITS
1134
1135 003076 012777 043576 175774 MOV #43576,RSCS1 ;SET DISK FUNCTION BITS
1136 003104 005077 175770 CLR RSCS1
1137 003110 022777 004200 175762 CMP #4200,RSCS1 ;IS THE READY BIT SET
1138 003116 001401 BEQ +4 ;YES
1139 003120 104001 HLT !CS1 ;RSCS1 SHOULD = 4200
1140
1141 ;*****
1142 ;TEST 5      TEST RSCS2
1143 ;*****
1144 003122 104400 TST5: SCOPE
1145
1146 003124 000005 RESET ;CLEAR WORLD
1147 003126 022777 000100 175746 CMP #100,RSCS2 ;DID THEY CLEAR?
1148 003134 001401 BEQ +4 ;YES
1149 003136 104200 HLT !CS2 ;NO
1150 003140 012777 021037 175734 MOV #21037,RSCS2 ;SET BITS 21017
1151 003146 022777 000137 175726 CMP #137,RSCS2 ;DID THESE BITS GET SET
1152 003154 001405 BEQ 15 ;YES
1153 003156 017700 175720 MOV RSCS2,BAD
1154 003162 012701 000137 MOV #137,GOOD ;WHAT CS2 SHOULD =
1155 003166 104000 HLT ;CS2 = BAD GOOD = CORRECT ANS
```

1156	003170	012777	020025	175704	1\$:	MOV	#20025, @RSCS2	;SET THESE BITS
1157	003176	022777	000125	175676		CMP	#125, @RSCS2	;DID THESE BITS GET SET
1158	003204	001401				BEQ	.+4	;YES
1159	003206	104200				HLT	!CS2	;NO CS2 SHOULD = 20125
1160	003210	012777	000012	175664		MOV	#12, @RSCS2	;LOAD THESE BITS
1161	003216	022777	000112	175656		CMP	#112, @RSCS2	;DID THESE BITS GET SET IN CS2
1162	003224	001401				BEQ	.+4	;YES
1163	003226	104200				HLT	!CS2	;BAD = CS2 GOOD = CORRECT ANS
1164	003230	012777	177777	175644		MOV	#-1, @RSCS2	;SET BITS
1165	003236	005077	175640			CLR	@RSCS2	;CLEAR THEM
1166	003242	022777	000100	175632		CMP	#100, @RSCS2	;DID CLEAR WORK
1167	003250	001401				BEQ	.+4	;YES
1168	003252	104200				HLT	!CS2	;R/W BITS DID NOT CLEAR
1169	003254	016777	175700	175620		MOV	UNNUM, @RSCS2	;GET UNIT #
1170	003262	104400						
1171					TST6:	SCOPE		
1172					;CAN WE	SET ALL	THE RSBA BITS	
1173	003264	012777	177777	175614		MOV	#177777, @RSBA	;SET THE BITS
1174	003272	022777	177776	175606		CMP	#177776, @RSBA	;DID THEY SET
1175	003300	001401				BEQ	.+4	;YES
1176	003302	104020				HLT	!BA	;BITS 17776 SHOULD BE SET
1177	003304	012777	125252	175574		MOV	#125252, @RSBA	;SET THESE BITS
1178	003312	022777	125252	175566		CMP	#125252, @RSBA	;ARE THEY =
1179	003320	001401				BEQ	.+4	;YES
1180	003322	104020				HLT	!BA	;SHOULD BE 125252
1181	003324	012777	052524	175554		MOV	#52524, @RSBA	;SET THESE BITS
1182	003332	022777	052524	175546		CMP	#52524, @RSBA	;ARE THEY =
1183	003340	001401				BEQ	.+4	;YES
1184	003342	104020				HLT	!BA	;SHOULD BE 52524
1185								
1186	003344	104400						
1187					TST7:	SCOPE		
1188					;FLOAT A 1	THROUGH RSBA		
1189	003346	012701	000002		FLOTBA:	MOV	#2, GOOD	;GET A 2
1190	003352	000241				CLC		;CLEAR CARRY
1191	003354	010177	175526		1\$:	MOV	GOOD, @RSBA	;FLOAT NUMBER
1192	003360	017700	175522			MOV	@RSBA, BAD	;GET BA
1193	003364	020100				CMP	GOOD, BAD	;COMPARE BA
1194	003366	001401				BEQ	.+4	;BA CORRECT
1195	003370	104000				HLT		;BAD=BA GOOD=CORRECT ANS
1196	003372	006101				ROL	GOOD	;ROTATE NUMBER
1197	003374	103367				BCC	1\$;LOOP TILL DONE


```

1198 003376 104400          TST10: SCOPE
1199
1200          ;CLEAR THE RSBA REGISTER
1201
1202 003400 012777 177777 175500      MOV    #177777, @RSBA    ;SET RSBA EQUAL TO ALL ONES
1203 003406 005077 175474              CLR    @RSBA
1204 003412 005777 175470              TST    @RSBA            ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1205 003416 001401                      BEQ    .+4              ;YES
1206 003420 104020                      HLT    !BA              ;NO
1207 003422 104400          TST11: SCOPE
1208
1209          ;CAN WE SET ALL BITS IN RSWC REGISTER
1210
1211 003424 012777 177777 175452      MOV    #177777, @RSWC    ;SET WC BITS
1212 003432 022777 177777 175444      CMP    #177777, @RSWC    ;ARE ALL BITS SET
1213 003440 001401                      BEQ    .+4              ;YES
1214 003442 104010                      HLT    !WC              ;NO
1215 003444 012777 125252 175432      MOV    #125252, @RSWC    ;SET THESE BITS
1216 003452 022777 125252 175424      CMP    #125252, @RSWC    ;ARE THEY =
1217 003460 001401                      BEQ    .+4              ;YES
1218 003462 104010                      HLT    !1C              ;SHOULD BE 125252
1219 003464 012777 052525 175412      MOV    #52525, @RSWC     ;SET THESE BITS
1220 003472 022777 052525 175404      CMP    #52525, @RSWC     ;ARE THEY =
1221 003500 001401                      BEQ    .+4              ;YES
1222 003502 104010                      HLT    !WC              ;SHOULD BE 152525
1223 003504 104400          TST12: SCOPE
1224
1225          ;FLOAT A 1 THROUGH RSWC
1226
1227 003506 012701 000001          FLOTWC: MOV    #1, GOOD    ;GET A 1
1228 003512 000241                      CLC                    ;CLEAR CARRY
1229 003514 010177 175364          1$:    MOV    GOOD, @RSWC ;FLOAT NUMBER
1230 003520 017700 175360          MOV    @RSWC, BAD      ;GET WC
1231 003524 020100                      CMP    GOOD, BAD        ;COMPARE WC
1232 003526 001401                      BEQ    .+4              ;WC CORRECT
1233 003530 104000                      HLT                    ;BAD=WC GOOD=CORRECT ANS
1234 003532 006101                      ROL                    ;ROTATE NUMBER
1235 003534 103367                      BCC    1$              ;LOOP TILL DONE

```

```

1236                                     :CLEAR THE WORD COUNT REGISTER
1237 003536 104400 TST13: SCOPE
1238
1239 003540 012777 177777 175336      MOV    #177777, @R5WC    ;SET R5WC REGISTER EQUAL TO ALL ONES
1240 003546 005077 175332              CLR    @R5WC
1241 003552 005777 175326              TST    @R5WC           ;DID ALL BITS GET CLEARED
1242 003556 001401                      BEQ    .+4             ;YES
1243 003560 104010                      HLT    !WC            ;NO
1244 003562 104400 TST14: SCOPE
1245
1246                                     ;CAN WE SET ALL THE BITS IN THE R5DA REGISTER.
1247
1248 003564 012777 177777 175316      MOV    #177777, @R5DA  ;SET ALL BITS
1249 003572 022777 177777 175310      CMP    #177777, @R5DA ;ARE THE BITS SET
1250 003600 001401                      BEQ    .+4             ;YES
1251 003602 104004                      HLT    !DA            ;NO
1252 003604 012777 125252 175276      MOV    #125252, @R5DA ;SET THESE BITS
1253 003612 022777 125252 175270      CMP    #125252, @R5DA ;ARE THEY =
1254 003620 001401                      BEQ    .+4             ;YES
1255 003622 104004                      HLT    !DA            ;SHOULD BE 125252
1256 003624 012777 052525 175256      MOV    #52525, @R5DA ;SET THESE BITS
1257 003632 022777 052525 175250      CMP    #52525, @R5DA ;ARE THEY =
1258 003640 001401                      BEQ    .+4             ;YES
1259 003642 104004                      HLT    !DA            ;SHOULD BE 52525
1260 003644 104400 TST15: SCOPE
1261
1262                                     ;FLOAT A 1 THROUGH R5DA
1263
1264 003646 012701 000001 FLOTDA: MOV    #1, GOOD ;GET A 1
1265 003652 000241              CLC                    ;CLEAR CARRY
1266 003654 010177 175230 1S:      MOV    GOOD, @R5DA    ;FLOAT NUMBER
1267 003660 017700 175224              MOV    @R5DA, BAD     ;GET DA
1268 003664 020100              CMP    GOOD, BAD      ;COMPARE DA
1269 003666 001401              BEQ    .+4             ;DA CORRECT
1270 003670 104000              HLT                    ;BAD=DA GOOD=CORRECT ANS
1271 003672 006101              ROL    GOOD           ;ROTATE NUMBER
1272 003674 103367              BCC    1$             ;LOOP TILL DONE

```



```

1273                                     ;CAN WE CLEAR THE RSDA REG.
1274 003676 104400 TST16: SCOPE
1275
1276 003700 012777 177777 175202      MOV    #177777, @RSDA ;SET RSDA TO ALL ONES
1277 003706 005077 175176             CLR    @RSDA          ;DID THEY SET
1278 003712 005777 175172             TST    @RSDA          ;TEST FOR ZERO RSDA
1279 003716 001401                     BEQ    .+4            ;YES
1280 003720 104004                     HLT    !DA           ;ANS SHOULD BE 0
1281 003722 104400 TST17: SCOPE
1282
1283                                     ;SET AND CLEAR THE RSER REG.
1284
1285 003724 012777 177017 175162      MOV    #177017, @RSER ;SET THESE BITS
1286 003732 022777 177017 175154      CMP    #177017, @RSER ;DID THEY SET
1287 003740 001401                     BEQ    .+4            ;YES
1288 003742 104002                     HLT    !ER           ;RSER SHOULD = 157017
1289 003744 112777 000001 175142      MOVB   #1, @RSER     ;A MOVB INST
1290 003752 022777 000001 175134      CMP    #1, @RSER     ;SHOULD MODIFY COMPLETE WD
1291 003760 001401                     BEQ    .+4            ;OK
1292 003762 104002                     HLT    !ER
1293
1294 003764 104400 TST20: SCOPE
1295
1296 003766 012777 052005 175120      MOV    #52005, @RSER ;SET THESE BITS
1297 003774 022777 052005 175112      CMP    #52005, @RSER ;DID THEY SET
1298 004002 001401                     BEQ    .+4            ;YES
1299 004004 104002                     HLT    !ER           ;ER SHOULD = 52005
1300 004006 104400 TST21: SCOPE
1301
1302 004010 012777 125012 175076      MOV    #125012, @RSER ;SET THESE BITS
1303 004016 022777 125012 175070      CMP    #125012, @RSER ;DID THEY SET
1304 004024 001401                     BEQ    .+4            ;YES
1305 004026 104002                     HLT    !ER           ;ER SHOULD = 105012

```

```

1306 004030 104400          TST22: SCOPE
1307
1308 004032 012777 177017 175054      MOV      #177017,RSER      ;SET THESE BITS
1309 004040 005077 175050              CLR      RSER             ;CLEAR THEM
1310 004044 005777 175044              TST      RSER            ;DID THEY CLEAR
1311 004050 001401              BEQ      +4              ;YES
1312 004052 104002              HLT      !ER             ;SHOULD = 0
1313 004054 104400          TST23: SCOPE
1314
1315                          ;SET AND CLEAR RSMR
1316
1317 004056 012777 000070 175040      MOV      #70,RSMR        ;SET THESE BITS
1318 004064 017767 175034 175122      MOV      RSMR,WORK       ;PUT INTO WORKABLE REG
1319 004072 042767 177700 175114      BIC      #177700,WORK    ;CLEAR JUNK
1320 004100 022767 000070 175106      CMP      #70,WORK        ;DID THEY SET
1321 004106 001401              BEQ      +4              ;YES
1322 004110 104220              HLT      !MR             ;SHOULD = 70
1323 004112 104400          TST24: SCOPE
1324
1325 004114 012777 000070 175002      MOV      #70,RSMR        ;SET BITS
1326 004122 005077 174776              CLR      RSMR            ;CLEAR THEM
1327 004126 032777 000077 174770      BIT      #77,RSMR        ;DID THEY CLEAR
1328 004134 001401              BEQ      +4              ;YES
1329 004136 104220              HLT      !MR             ;BITS (77) SHOULD = 0
1330 004140 104400          TST25: SCOPE
1331
1332 004142 012777 000050 174754      MOV      #50,RSMR        ;SET BITS
1333 004150 017767 174750 175036      MOV      RSMR,WORK       ;PUT IN WORKABLE REG
1334 004156 042767 177700 175030      BIC      #177700,WORK    ;CLEAR JUNK
1335 004164 022767 000050 175022      CMP      #50,WORK        ;DID THESE BITS SET
1336 004172 001401              BEQ      +4              ;YES
1337 004174 104220              HLT      !MR             ;BITS (50) SHOULD BE SET
1338 004176 104400          TST26: SCOPE
1339
1340 004200 012777 000020 174716      MOV      #20,RSMR        ;SET BITS
1341 004206 017767 174712 175000      MOV      RSMR,WORK       ;PUT INTO WORKABLE REG
1342 004214 042767 177700 174772      BIC      #177700,WORK    ;CLEAR JUNK
1343 004222 022767 000020 174764      CMP      #20,WORK        ;DID THEY SET
1344 004230 001401              BEQ      +4              ;YES
1345 004232 104220              HLT      !MR             ;MR SHOULD AT LEAST HAVE A (21)

```



```

1346 ;*****
1347 ;TEST 27 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1348 ;*****
1349 004234 104400 TST27: SCOPE
1350
1351 004236 104414 BITST: CLRDK ;CLEAR ALL RS REG
1352 004240 012777 003566 174632 MOV #3566,RS1 ;LOAD CS1
1353 004246 112777 000005 174660 MOVB #5,RS1B ;LOAD BIT
1354 004254 022777 004766 174616 CMP #4766,RS1 ;DID IT LOAD?
1355 004262 001401 BEQ +4 ;YES
1356 004264 104001 HLT !CS1
1357 004266 112777 000032 174604 MOVB #32,RS1
1358 004274 022777 004632 174576 CMP #4632,RS1
1359 004302 001401 BEQ +4
1360 004304 104001 HLT !CS1 ;CS1 SHOULD = 4632
1361
1362 004306 104400 TST30: SCOPE
1363
1364 004310 016777 174644 174564 BITCS2: MOV UNNUM,RS2 ;LOAD UNIT NUMBER
1365 004316 052777 177400 174556 BIS #177400,RS2 ;LOAD ALL BITS
1366 004324 105077 174606 CLRB RS2B ;CLR UPPER BYTE
1367 004330 016701 174624 MOV UNNUM,GOOD ;GET UNIT NO.
1368 004334 052701 000100 BIS #100,GOOD ;SET OR BIT
1369 004340 017700 174536 MOV RS2,BAD ;GET CS2
1370 004344 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1371 004346 001401 BEQ +4 ;YES
1372 004350 104000 HLT ;LOAD BYTE DID NOT WORK
1373
1374 004352 104400 TST31: SCOPE
1375
1376 004354 012777 025252 174522 BITWC: MOV #25252,RSWC ;LOAD WC
1377 004362 112777 000377 174550 MOVB #377,RSWCB ;LOAD BIT
1378 004370 022777 177652 174506 CMP #177652,RSWC ;DID IT LOAD?
1379 004376 001401 BEQ +4 ;YES
1380 004400 104010 HLT !WC ;NO WC SHOULD =177652
1381 004402 112777 000123 174474 MOVB #123,RSWC
1382 004410 022777 177523 174466 CMP #177523,RSWC
1383 004416 001401 BEQ +4
1384 004420 104010 HLT !WC ;WC SHOULD = 177523
1385
1386 004422 104400 TST32: SCOPE
1387
1388 004424 012777 025252 174454 BITBA: MOV #25252,RSBA ;LOAD DA
1389 004432 112777 000377 174502 MOVB #377,RSBAB ;LOAD BIT
1390 004440 022777 177652 174440 CMP #177652,RSBA ;DID IT LOAD?
1391 004446 001401 BEQ +4 ;YES
1392 004450 104020 HLT !BA ;DA SHOULD =177652
1393 004452 112777 000125 174426 MOVB #125,RSBA
1394 004460 022777 177524 174420 CMP #177524,RSBA
1395 004466 001401 BEQ +4
1396 004470 104020 HLT !BA ;BA SHOULD = 177525
1397 004472 104414 CLRDK ;CLEAR ALL RS REG

```

```

1398
1399
1400
1401 004474 104400
1402 004476 104414
1403 004500 005077 174416
1404 004504 012777 177777 174410
1405 004512 012767 002000 174474
1406 004520 012701 000300
1407 004524 056701 174430
1408 004530 017700 174346
1409 004534 020100
1410 004536 001404
1411 004540 005367 174450
1412 004544 001371
1413 004546 104200
1414 004550 005001
1415 004552 017700 174344
1416 004556 020100
1417 004560 001401
1418 004562 104000
1419 004564 012701 177777
1420 004570 017700 174326
1421 004574 020100
1422 004576 001401
1423 004600 104000

*****
:TEST 33 LOAD RSDB WITH ALL ONES AND ALL ZEROS
*****
TST33: SCOPE
ZERONE: CLRDK
CLR JRSDB ;CLEAR ALL RS REG
MOV #177777, JRSDB ;LOAD DB WITH ALL 0
MOV #2000, WORK ;LOAD DB WITH ALL ONES
MOV #300, GOOD ;TIME OUT ROUTINE
BIS UNNUM, GOOD ;GET CORRECT FOR CS2
MOV JRSDB, BAD ;GET CS2
CMP GOOD, BAD ;IS IT CORRECT?
BEQ 3$ ;YES
DEC WORK ;TO WAIT FOR OR
BNE 2$ ;TO SET
HLT !CS2 ;OR SHOULD BE SET
CLR GOOD
MOV JRSDB, BAD ;LOAD BAD WITH DB
CMP GOOD, BAD ;IS BAD CORRECT
BEQ .+4 ;YES
HLT ;COULD NOT FLOAT 0 THROUGH DB
MOV #-1, GOOD ;LOAD GOOD WITH ANS
MOV JRSDB, BAD ;GET DATA FROM DB
CMP GOOD, BAD ;IS DB CORRECT
BEQ .+4 ;YES
HLT ;BAD SHOULD = 177777

```



```

1424 ;TEST INTERRUPT IN THE RH11
1425 ;BY MOVING 300 INTO RHCS1
1426 ;*****
1427 ;TEST 34 TEST INTERRUPT IN RH11
1428 ;*****
1429 004602 104400 TST34: SCOPE
1430 004604 104414 INT: CLRDK ;CLEAR ALL ERRORS
1431 004606 012777 004660 174314 MOV #PGTRAP,ARVVEC ;SET UP VECTOR
1432 004614 012777 000340 174310 MOV #340,ARVVCPS ;SET TRAP PS
1433 004622 012737 000200 177776 MOV #200,ARPS ;SET PS AT PRIORITY 4
1434 004630 012777 000300 174242 MOV #300,ARSCS1 ;THIS SHOULD CAUSE A TRAP
1435 004636 012767 000500 174350 MOV #500,WORK ;SETUP LOOP
1436 004644 005367 174344 IS: DEC WORK ;DEC LOOP SHOULD
1437 004650 001375 BNE IS ;INTERRUPT BEFORE LOOP IS DONE
1438 004652 104001 HLT !CS1 ;SHOULD NEVER GET HERE
1439 004654 000167 000014 JMP INTDON ;GET OUT
1440 004660 022626 PGTRAP: CMP (6)+,(6)+ ;TRAP OK
1441 004662 022777 004200 174210 CMP #4200,ARSCS1 ;DID IE CLEAR?
1442 004670 001401 BEQ +4 ;YES
1443 004672 104001 HLT !CS1 ;IE SHOULD BE CLEARED
1444 004674 INTDON:

```

```

1445 :*****
1446 :TEST 35 MAINTENANCE TIMING TEST
1447 :*****
1448 004674 104400 TST35: SCOPE
1449
1450 :MODULE TESTED G092
1451 :THE FOLLOWING TEST ON THE RS04 DISK IS A SINGLE-STEPPED
1452 :MAINTENANCE MODE TEST ON THE RS04 TIMING LOGIC. THE ACTUAL
1453 :DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE RESISTER--I.E.
1454 :THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE
1455 :TIMING LOGIC. WE ARE TESTING THE ENTIRE TIMING TRACK LOGIC, INCLUDING, INDEX
1456 :PULSE FUNCTION, RESYNC AREA, SECTOR COUNTERS, ETC.
1457
1458 :PUT DRIVE IN MAINTENANCE MODE
1459 004676 104414 MRTIME: CLRDK ;CLEAR DRIVE REGISTERS
1460 004700 052767 001040 174260 BIS #1040,ONCEE ;SET CLK CNT
1461 004706 104430 MRIND ;SEND INDEX PULSE TO MR REG
1462 004710 104420 MRCK ;CHECK MAINTENANCE REG FOR
1463 004712 022701 22701 ;22701
1464 004714 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1465 ;BY SENDING 2 CLOCK PULSES
1466 004716 104430 MRIND ;SEND MAINT INDEX PULSE
1467
1468 004720 104420 MRCK ;CHECK MAINT REG TO
1469 004722 022701 22701 ;EQUAL 22701
1470 004724 104000 HLT ;MR=BAD GOOD=CORRECTIONS
1471 ;COULD NOT INITIALIZE MR REG
1472 ;INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG
1473
1474 004726 005777 174166 TST RSLA ;IS RSLA CLEARED
1475 004732 001401 BEQ +4 ;YES
1476 004734 104224 HLT !MR!LA ;RSLA SHOULD BE CLEARED
1477 ;WITH THE INDEX PULSE
1478
1479 :PERFORM MAINTENANCE CLOCK OPERATION 1024 TIMES TO
1480 :PROVIDE CLOCK TO STEP TIMING THRU RESYNC PERIOD.
1481 :IF SECTOR PULSE IS ASSERTED DURING THIS LOOP
1482 :CHECK SECTOR BOUNDARY COUNTER AND E12
1483
1484 004736 012767 001000 174236 MRTIM1: MOV #512.,REPT
1485 004744 104422 MRCLK ;CLOCK MAINT REG WITH AN 11 AND A 1
1486 004746 104420 MRCK ;CHECK MR REG TO
1487 004750 072701 72701 ;EQUAL 72701
1488 004752 104000 HLT ;MR = BAD, GOOD = CORRECT ANS
1489 004754 104422 MRCLK ;CLOCK MR
1490 004756 104420 MRCK ;CHECK MR TO
1491 004760 022701 22701 ;EQUAL 22701
1492 004762 104000 HLT ;BAD=MR REG GOOD=CORRECTIONS
1493 004764 005367 174212 DEC REPT ;IS THE LOOP DONE YET?
1494 004770 001365 BNE MRTIM1 ;NO-LOOP

```


E04

MAINDEC-11-DERSD-8
DERSDB.P11 TST35

RS11-R504 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE TIMING TEST

MACY11 27(732) 04-OCT-76 13:11 PAGE 44

```
1495 ;AFTER ONE MORE CLOCK SECTOR PULSE SHOULD BE ASSERTED
1496 ;IF NOT, CHECK SECTOR BOUNDARY COUNTER, SECTOR BOUNDARY FF (E21) AND E12
1497
1498 004772 104422 MRCLK ;CLOCK MAINT REG WITH A 11 AND A 1
1499 004774 104420 MRCK ;CHECK MR REG TO
1500 004776 072301 72301 ;EQUAL 72301
1501 005000 104000 HLT ;MR=BAD GOOD=CORRECTIONS
1502 005002 104422 MRCLK ;CLOCK MR WITH 11 AND A 1
1503 005004 104420 MRCK ;CHECK MAINT REG
1504 005006 022301 22301 ;TO EQUAL 22301
1505 005010 104000 HLT ;MR=BAD GOOD-CORRECT ANS
1506 005012 005777 174102 TST JRSLA ;DOES LOOK AHEAD REG=0
1507 005016 001401 BEQ MRT2 ;YES-CONT
1508 005020 104224 HLT !MR!LA ;LOOK AHEAD REG SHOULD=0
1509 ;PERFORM MAINTENANCE CLOCK OPERATION 80 TIMES TO PROVIDE
1510 ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA
1511
1512 005022 005002 MRT2: CLR R2 ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
1513 005024 012767 000050 174150 MOV #40.,REPT ;80 CLOCKS TO STEP THRU PRE-AMBLE
1514 005032 104422 MRT2A: MRCLK ;CLOCK MR WITH A 11 AND A 1
1515 005034 104420 MRCK ;CHECK MAINT REG
1516 005036 073701 73701 ;EQUAL 73701
1517 005040 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1518 005042 104422 MRCLK ;CLOCK MR REG
1519 005044 104420 MRCK ;CHECK MR REG
1520 005046 023701 23701 ;TO EQUAL 23701
1521 005050 104000 HLT ;MR = BAD GOOD = CORRECTANS
1522 005052 005367 174124 DEC REPT ;REPEAT
1523 005056 001365 BNE MRT2A ;LOOP 40 TIMES
1524
1525 ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
1526 005060 012767 002200 174114 MOV #9.*128.,REPT ;18 CLOCKS PER DATA WORD
1527 005066 104422 MRT2B: MRCLK ;CLOCK MR WITH A 11 AND A 1
1528 005070 104420 MRCK ;CHECK MAINT REG
1529 005072 073701 73701 ;TO EQUAL 73701
1530 005074 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1531 005076 104422 MRCLK ;CLOCK MR REG
1532 005100 104420 MRCK ;CHECK MR REG
1533 005102 023701 23701 ;TO EQUAL 23701
1534 005104 104000 HLT ;MR=BAD GOOD=CORRECTANS
1535 005106 005367 174070 DEC REPT ;REPEAT
1536 005112 001365 BNE MRT2B ;LOOP
```

```

1537 ;SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
1538 ;AND THE DEAD BAND ON THE SECTOR
1539
1540 005114 012767 000214 174060 MRT2C: MOV #140.,REPT ;AMOUNT OF CLOCKS TO END OF SECTOR
1541 005122 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1542 005124 104420 MRCK ;CHECK MAINT REG
1543 005126 073701 73701 ;TO EQUAL 73701
1544 005130 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1545 005132 104422 MRCLK ;CLOCK MR REG
1546 005134 104420 MRCK ;CHECK MAINT REG
1547 005136 023701 23701 ;TO EQUAL 23701
1548 005140 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1549 005142 005367 174034 DEC REPT ;REPEAT
1550 005146 001365 BNE MRT2C ;LOOP
1551 005150 104422 MRCLK ;CLOCK MR REG
1552 005152 104420 MRCK ;CHECK MR REG
1553 005154 073701 73701 ;TO EQUAL 73701
1554 005156 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1555 ;ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
1556 ;IF NOT, CHECK E16-6
1557
1558 005160 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1559 005162 104420 MRCK ;MAINT REG SHOULD
1560 005164 023701 23701 ;EQUAL 22301
1561 005166 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1562 005170 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1563 005172 104420 MRCK ;MAINT REG
1564 005174 072301 72301 ;SHOULD EQUAL 72301
1565 005176 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1566
1567 ;LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)
1568
1569 005200 022777 000002 173720 CMP #2,RSDDT ;INTERLEAVED?
1570 005206 001403 BEQ 3$ ;NO
1571 005210 062702 004000 ADD #4000,R2 ;YES
1572 005214 000402 BR 2$ ;CONT
1573 005216 062702 000100 3$: ADD #100,R2 ;INCREMENT SECTOR COMPARE
1574 005222 020277 173672 2$: CMP R2,RSLSA ;LA REG SHOULD=100
1575 005226 001401 BEQ 1$ ;LA IS CORRECT
1576 005230 104224 HLT !MR!LA ;LA SHOULD=100

```



```

1577 ;REPEAT NEXT STEPS 62 TIMES. LOOK-AHEAD REGISTER SHOULD INCREMENT
1578 ;TO SHOW NEXT SECTOR. CHECKS FOR ALL SECTORS. IF DRIVE IS NOT
1579 ;INTERLEAVED, LA = 200,300, ETC. IF DRIVE IS INTERLEAVED,
1580 ;LA = 100, 4100, 200, 4200 ETC. SEE SERVICE MANUAL FOR DETAILS.
1581
1582 005232 012767 000076 173744 1$: MOV #62.,REPT1
1583 005240 012767 005152 173734 MRT3: MOV #2666.,REPT
1584 005246 104422 3$: MRCLK
1585 005250 005367 173726 DEC REPT
1586 005254 001374 BNE 3$
1587 005256 104422 MRCLK
1588 005260 104420 MRCK
1589 005262 022701 22701
1590 005264 104000 HLT
1591 005266 104422 MRCLK
1592 005270 104420 MRCK
1593 005272 072301 72301
1594 005274 104000 HLT
1595 005276 022777 000002 173622 CMP #2,RSDDT
1596 005304 001420 BEQ 6$
1597 005306 032767 001000 173652 BIT #BIT9,ONCEE
1598 005314 001406 BEQ 4$
1599 005316 042767 001000 173642 BIC #BIT9,ONCEE
1600 005324 162702 004000 SUB #4000,R2
1601 005330 000406 BR 6$
1602 005332 052767 001000 173626 4$: BIS #BIT9,ONCEE
1603 005340 062702 004000 ADD #4000,R2
1604 005344 000402 BR 5$
1605 005346 062702 000100 6$: ADD #100,R2
1606 005352 017700 173542 5$: MOV RSLA,BAD
1607 005356 010201 MOV R2,GOOD
1608 005360 020100 CMP GOOD,BAD
1609 005362 001401 BEQ 1$
1610 005364 104000 HLT
1611
1612 005366 005367 173612 1$: DEC REPT1
1613 005372 001322 BNE MRT3
1614 005374 012767 005152 173600 2$: MOV #2666.,REPT
1615 005402 104422 MRCLK
1616 005404 005367 173572 DEC REPT
1617 005410 001374 BNE 2$
1618 005412 017700 173502 MOV RSLA,BAD
1619 005416 012701 007777 MOV #7777,GOOD
1620 005422 020100 CMP GOOD,BAD
1621 005424 001401 BEQ .+4
1622 005426 104000 HLT

```

```

;CLOCK MR WITH A 11 AND A 1
;STEP THROUGH
;SECTOR
;CLOCK MR WITH A 11 AND A 1
;MAINT REG
;SHOULD EQUAL 22701
;MR=BAD GOOD=CORRECT ANS
;1 MORE CLK, SAME SECTOR PULSE
;MAINT REG SHOULD
;EQUAL 72301
;MR=BAD GOOD=CORRECT ANS
;DRIVE INTERLEAVED?
;NO
;DO I ADD 4000
;OR SUBTRACT IT FROM WHAT I EXPECT TO
;FIND IN RSLA

```

```

;INCREMENT SECTOR COMPARE
;LA REG SHOULD HAVE INCREMENTED TO NEXT SECTOR
;GET CORRECT ANS FOR RSLA
;COMPARE FOR CORRECT ANS
;RSLA IS GOOD
;RSLA=BAD GOOD=CORRECT ANS

```

```

;REPEAT 62
;TIMES
;COUNT FOR LAST SECTOR
;CLOCK
;THRU
;LAST SECTOR
;GET CONTENTS OF RSLA
;GET CORRECT ANS
;DOES RSLA EQUAL 7777
;YES
;BAD=RSLA GOOD=CORRECT ANS

```

-897

```

1623 ;*****
1624 ;TEST 36 SECTOR FRACTION TEST
1625 ;*****
1626 005430 104400 TST36: SCOPE
1627 ;MODULE TESTED G092
1628 ;CLOCK THROUGH AN ENTIRE TRACK IN MAINT MODE WHILE
1629 ;CHECKING FOR THE PROPER OPERATION OF THE LOOK-AHEAD REGISTER AND
1630 ;THE SECTOR FRACTION COUNTER. WHEN THE LAST WORD IS BEING TRANSFERRED,
1631 ;SECTOR AND FRACTION IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK --
1632 ;HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL CLEAR THE
1633 ;FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA SHOULD INDICATE 7700 ON
1634 ;ANOTHER MAINTENANCE CLOCK.
1635
1636 005432 104414 MRT4: CLRDK ;CLEAR DRIVE REGISTERS
1637 005434 052767 000040 173524 BIS #40,ONCEE ;SET FLAG BITS
1638 005442 042767 003000 173516 BIC #3000,ONCEE
1639 005450 005067 173520 CLR MCCNT ;CLEAR MAINT CLOCK COUNTER
1640 005454 005002 CLR R2 ;CLEAR R2 FOR SECTOR COUNTER
1641 005456 104430 MRIND ;SEND INDEX PULSE TO MR REG
1642 005460 104420 MRCK ;CHECK MR REG
1643 005462 022701 22701 ;TO EQUAL 22701
1644 005464 104424 MRINT ;INIT MAINT MODE
1645 005466 104430 MRIND ;ISSUE A MAINT INDEX PULSE
1646 ;TO CLEAR THE DRIVE
1647 005470 104420 MRCK ;CHECK MAINT REG
1648 005472 022701 22701 ;TO EQUAL 22701
1649 005474 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1650
1651 ;ISSUE 1024 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA
1652
1653 005476 012767 001000 173476 MRT4A: MOV #512.,REPT ;COUNT TO STEP THRU RESYNC AREA
1654 005504 104422 MRCLK ;CLOCK THROUGH RESYNC
1655 005506 104420 MRCK ;CHECK MAINT REG
1656 005510 072701 72701 ;TO EQUAL 72701
1657 005512 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1658 005514 005777 173400 TST @RSLA ;IS RSLA=TO 0
1659 005520 001401 BEQ +4 ;YES
1660 005522 104204 HLT !LA ;RSLA SHOULD=0 DURING RESPONSE
1661 005524 104422 MRCLK ;CLOCK MR REG
1662 005526 104420 MRCK ;CHECK MR REG
1663 005530 022701 22701 ;TO EQUAL 22701
1664 005532 104000 HLT ;BAD=MR GOOD=CORRECT ANS
1665 005534 005777 173360 TST @RSLA ;IS RSLA=TO 0
1666 005540 001401 BEQ +4 ;YES
1667 005542 104204 HLT !LA ;RSLA SHOULD=0 DURING RESPONSE
1668 005544 005367 173432 DEC REPT ;LOOP THROUGH
1669 005550 001355 BNE MRT4A ;RESYNC AREA
1670
1671 ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE
1672
1673 005552 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1674 005554 104420 MRCK ;CHECK MAINT REG FOR SECTOR PULSE
1675 005556 072301 72301 ;MR SHOULD=72301
1676 005560 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```



```

1677 005562 104422          MRT4B: MRCLK          ;CLOCK MR REG WITH A 11 AND A 1
1678 005564 104420          MRCK          ;CHECK MAINT REG
1679 005566 022301          22301        ;TO EQUAL 22301
1680 005570 104000          HLT          ;MR=BAD GOOD=CORRECT ANS
1681
1682          ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)
1683
1684 005572 017700 173322    MOV          @RSLA,BAD ;GET RSLA
1685 005576 010201          MOV          R2,GOOD  ;GET CORRECT ANS
1686 005600 020100          CMP          GOOD,BAD ;IS THE RSLA REG CORRECT
1687 005602 001401          BEQ         1$        ;YES
1688 005604 104000          HLT          ;RSLA=BAD GOOD=CORRECTANS
1689
1690          ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
1691          ;AREA WHILE CHECKING THE SECTOR FRACTION
1692
1693 005606 012767 000244 173366 1$: MOV          #164.,REPT ;FOR FIRST FRACTION CHANGE
1694 005614 104422          MRT4C: MRCLK          ;CLOCK MR REG WITH A 11 AND A 1
1695 005616 017700 173276    MOV          @RSLA,BAD ;GET RSLA
1696 005622 010201          MOV          R2,GOOD  ;GET CORRECT ANS
1697 005624 020001          CMP          BAD,GOOD ;IS RSLA CORRECT
1698 005626 001401          BEQ         1$        ;YES
1699 005630 104000          HLT          ;BAD=RSLA GOOD=CORRECT ANS
1700 005632 005367 173344    1$: DEC          REPT  ;LOOP ON
1701 005636 001366          BNE          MRT4C    ;PREAMBLE AREA
1702
1703          ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1704
1705 005640 104422          MRCLK          ;CLOCK MR WITH A 11 AND A 1
1706 005642 005202          INC          R2        ;COUNT THE FRACTION
1707 005644 017700 173250    MOV          @RSLA,BAD ;GET RSLA
1708 005650 010201          MOV          R2,GOOD  ;GET CORRECT ANS
1709 005652 020001          CMP          BAD,GOOD ;IS RSLA CORRECT?
1710 005654 001401          BEQ         2$        ;YES
1711 005656 104000          HLT          ;RSLA=BAD GOOD=CORRECT ANS
1712
1713          ;FIRST FRACTION CHANGES AFTER 164 MAINT. CLKS, THE REST
1714          ;CHANGE AFTER 40 MAINTENANCE CLOCKS
1715
1716 005660 012767 000076 173314 2$: MOV          #62.,REPT ;COUNT FOR WORDS IN A SECTOR
1717 005666 012767 000047 173310 MRT4D: MOV          #39.,REPT1 ;COUNT FOR SECT FRACT TO CHANGE
1718 005674 104422          MRT4E: MRCLK          ;CLOCK MR WITH A 11 AND A 1
1719 005676 017700 173216    MOV          @RSLA,BAD ;GET RSLA
1720 005702 010201          MOV          R2,GOOD  ;GET CORRECT ANS
1721 005704 020100          CMP          GOOD,BAD ;IS RSLA CORRECT?
1722 005706 001401          BEQ         1$        ;YES
1723 005710 104000          HLT          ;RSLA=BAD GOOD=CORRECT ANS
1724 005712 005367 173266    1$: DEC          REPT1 ;LOOP
1725 005716 001366          BNE          MRT4E

```

```

1726                                     ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1727
1728 005720 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1729 005722 022702 007777 CMP #7777,R2 ;AT THE LAST SECTOR-LAST FRACTION?
1730 005726 001472 BEQ MRT4F ;YES, FINISH THE SECTOR
1731 005730 005202 INC R2 ;NO, ADD 1 TO FRACTION
1732 005732 017700 173162 4$: MOV @RSLA,BAD ;GET RSLA
1733 005736 022777 000002 173162 CMP #2,@RSDT ;IS THIS DIRVE INTERLEAVED?
1734 005744 001431 BEQ 12$ ;NO
1735 005746 032767 002000 173212 BIT #DIT10,ONCEE ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
1736 005754 001425 BEQ 12$ ;NO
1737 ;RSLA NOW POINTS TO NEXT INTERLEAVED SECTOR BIT 9 IN ONCEE
1738 ;INDICATES WHETHER RSLA SHOULD NOW BE BETWEEN
1739 ;0000-3700(1) OR 4000-7700(0).
1740 005756 032767 001000 173202 BIT #BIT9,ONCEE ;SHOULD RSLA BE BETWEEN 0-3700?
1741 005764 001004 BNE 9$ ;YES
1742 005766 052767 001000 173172 BIS #BIT9,ONCEE ;SET FOR NEXT PASS
1743 005774 000406 BR 10$
1744 005776 042767 001000 173162 9$: BIC #BIT9,ONCEE ;CLEAR FOR NEXT PASS
1745 006004 042702 004000 BIC #4000,R2 ;MAKE EXPECTED RSLA LESS THAN 4000
1746 006010 000404 BR 5$
1747 006012 062702 004000 10$: ADD #4000,R2 ;COMPENSATE FOR INTERLEAVING
1748 006016 162702 000100 SUB #100,R2
1749 006022 042767 002000 173136 5$: BIC #BIT10,ONCEE ;CLEAR FLAG FOR NEXT SECTOR
1750 006030 010201 12$: MOV R2,GOOD ;GET CORRECT ANSWER FOR RSLA
1751 006032 020100 CMP GOOD,BAD ;IS RSLA CORRECT
1752 006034 001401 BEQ 2$ ;YES
1753 006036 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
1754 006040 005367 173136 2$: DEC REPT ;HAS SECTOR FRACTION REACHED 77?
1755 006044 001310 BNE MRT4D ;NO
1756
1757 ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT
1758
1759 006046 010203 11$: MOV R2,R3
1760 006050 042703 177700 BIC #177700,R3 ;CHECK SECTOR FRACTION
1761 006054 022703 000077 CMP #77,R3 ;END OF SECTOR?
1762 006060 001402 BEQ 3$ ;YES
1763 006062 000167 177474 JMP MRT4B ;NO, BEGINNING OF NEXT
1764 006066 012767 000025 173110 3$: MOV #21.,REPT1 ;SETUP LOOP TO FINISH
1765 006074 012767 000001 173100 MOV #1,REPT ;THIS SECTOR
1766 006102 052767 002000 173056 BIS #BIT10,ONCEE ;REPT HAS GONE TO ZERO FOR THIS SECTOR
1767 006110 000167 177560 JMP MRT4E ;LOOP
1768
1769 006114 012767 000021 173060 MRT4F: MOV #17.,REPT
1770 006122 104422 1$: MRCLK ;CLOCK MR WITH A 11 AND A 1
1771 006124 017700 172770 MOV @RSLA,BAD ;GET RSLA
1772 006130 010201 MOV R2,GOOD ;R2 SHOULD=7777
1773 006132 020100 CMP GOOD,BAD ;IS RSLA CORRECT-END OF DISK?
1774 006134 001401 BEQ 2$ ;YES
1775 006136 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS (7777)
1776 006140 005367 173036 2$: DEC REPT ;FINISH
1777 006144 001366 BNE 1$ ;LOOP

```


1778			
1779			
1780			
1781	006146	104422	
1782	006150	017700	172744
1783	006154	012701	007700
1784	006160	020100	
1785	006162	001401	
1786	006164	104000	
1787	006166	104430	
1788			
1789	006170	017700	172724
1790	006174	005001	
1791	006176	020100	
1792	006200	001401	
1793	006202	104000	
1794	006204	104420	
1795	006206	022701	
1796	006210	104000	

:SECTOR AND FRACTION IS = TO 7777 TO INDICATE LAST WORD ON THIS TRACK
:RSLA SHOULD EQUAL 7700 ON ANOTHER MAINT CLOCK.

MRT4G:	MRCLK		:CLOCK MR WITH A 11 AND A 1
	MOV	2RSLA,BAD	:GET RSLA
	MOV	#7700,GOOD	:GET CORRECT ANS
	CMP	GOOD,BAD	:IS RSLA CORRECT?
	BEQ	1S	:YES
	HLT		:RSLA=BAD GOOD=CORRECT ANS
1S:	MRIND		:ISSUE AN INDEX PULSE TO
			:CLEAR THE DRIVE
	MOV	2RSLA,BAD	:GET RSLA
	CLR	GOOD	:GET CORRECT ANS
	CMP	GOOD,BAD	:IS RSLA CORRECT?
	BEQ	2S	:YES
	HLT		:RSLA=BAD GOOD=CORRECT ANS
2S:	MRCK		:CHECK MR REG
	22701		:TO EQUAL 22701
	HLT		:MR=BAD GOOD=CORRECT ANS

```

1797
1798
1799
1800 006212 104400
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812 006214 104414
1813 006216 042767 000040 172742
1814 006224 032767 000002 172734
1815 006232 001002
1816 006234 012702 000003
1817
1818
1819 006240 104416
1820 006242 104420
1821 006244 022701
1822 006246 104424
1823
1824
1825
1826 006250 104430
1827 006252 010277 172622
1828 006256 017700 172630
1829 006262 012701 150600
1830 006266 020100
1831 006270 001440
1832 006272 104402 006276
1833 006346 010267 172642
1834 006352 016746 172636
1835 006356 104406
1836 006360 052767 000002 172600
1837 006366 104000
1838 006370 104040
1839
1840 006372 042767 000002 172566 1$:
1841 006400 017700 172510
1842 006404 012701 000001
1843 006410 020100
1844 006412 001404
1845 006414 052767 000002 172544
1846 006422 104000
1847 006424 042767 000002 172534 2$:

```

```

;*****
;TEST 37 ILLEGAL FUNCTION TEST
;*****
TST37: SCOPE

;MODULE TESTED M7759, M7770
;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.
;ILLEGAL FUNCTIONS ARE DETECTED ON M7759 BY E20-8

MRILF: CLDK ;CLEAR ALL THE DRIVE REGISTERS
        BIC #BITS,ONCEE ;CLEAR CLOCK CNT FLAG
        BIT #BIT1,ONCEE ;WAS THERE AN ERROR
        BNE MRLF1 ;YES DO NOT CHANGE "ILF" CODE
        MOV #3,R2 ;SETUP FIRST "ILF" CODE
;PUT DRIVE IN MAINTENANCE MODE

MRLF1: MRDMD ;PUT DRIVE INTO MAINT MODE
        MRCK ;CHECK MR REG TO
        22701 ;EQUAL 22701
        MRINT ;INIT MAINT MODE (CLEAR MRSP)

;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE

MRLF2: MRIND
        MOV R2,RSCS1 ;SEND "ILF" WITH THE "GO" BIT
        MOV RSDS,BAD ;GET DRIVE STATUS REG
        MOV #150600,GOOD ;GET CORRECT ANS
        CMP GOOD,BAD ;IS RSDS CORRECT?
        BEQ 1$ ;YES
        TYPE .ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE="
        MOV R2,WORK ;GET FUNCTION CODE
        MOV WORK,-(6) ;PUT WORK ON STACK
        TYPES ;TYPE STACK IN OCTAL - SUPRESS
        BIS #BIT1,ONCEE ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
        HLT ;RSDS=BAD GOOD=CORRECT ANS
        HLT !DS

1$: BIC #BIT1,ONCEE ;CLEAR ERROR FLAG
    MOV RRSER,BAD ;GET RSER
    MOV #1,GOOD ;GET CORRECT ANS
    CMP GOOD,BAD ;DID "ILF" SET IN RSER
    BEQ 2$ ;YES
    BIS #BIT1,ONCEE ;SET ERROR BIT
    HLT ;RSER=BAD GOOD=CORRECT ANS
    BIC #BIT1,ONCEE ;CLEAR ERROR FLAG

```



```

1848                                     ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
1849 006432 104414 MRCILF: CLRDK                                     ;CLEAR ERRORS
1850 006434 017700 172452 MOV      @RSDS,BAD                       ;GET RSDS REG
1851 006440 012701 010600 MOV      #10600,GOOD                    ;GET CORRECT ANS
1852 006444 020100 CMP      GOOD,BAD                               ;DID "ATA" AND "ERR" CLEAR IN RSDS?
1853 006446 001435 BEQ      1$                                       ;YES
1854 006450 104402 006454 TYPE     ,.+2                               ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
1855 006532 052767 000002 172426 BIS      #BIT1,ONCEE
1856 006540 104000 HLT
1857 006542 042767 000002 172416 1$: BIC      #BIT1,ONCEE          ;RSDS=BAD GOOD=CORRECT ANS
1858 006550 017700 172340 MOV      @RSER,BAD                       ;CLEAR ERROR FLAG
1859 006554 005001 CLR      GOOD                                       ;GET RSER
1860 006556 020100 CMP      GOOD,BAD                               ;GET CORRECT ANS
1861 006560 001431 BEQ      2$                                       ;DID ILF CLEAR IN RSER
1862 006562 052767 000002 172376 BIS      #BIT1,ONCEE          ;YES
1863 006570 104402 006574 TYPE     ,.+2                               ;SET ERROR BIT
1864 006642 104000 HLT                                       ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
1865 006644 042767 000002 172314 2$: BIC      #BIT1,ONCEE          ;RSER=BAD GOOD=CORRECT ANS
1866                                     ;GET NEXT ILLEGAL FUNCTION COE
1867
1868 006652 062702 000002 MRLF3: ADD      #2,R2                               ;UPDATE ILF
1869 006656 022702 000011 CMP      #11,R2                               ;IS THIS A ILF CODE
1870 006662 001773 BEQ      MRLF3                                       ;NO-UPDATE IT
1871 006664 022702 000021 CMP      #21,R2
1872 006670 001770 BEQ      MRLF3
1873 006672 022702 000031 CMP      #31,R2
1874 006676 001765 BEQ      MRLF3
1875 006700 022702 000051 CMP      #51,R2
1876 006704 001762 BEQ      MRLF3
1877 006706 022702 000061 CMP      #61,R2
1878 006712 001757 BEQ      MRLF3
1879 006714 022702 000071 CMP      #71,R2
1880 006720 001754 BEQ      MRLF3
1881 006722 022702 000101 CMP      #101,R2
1882 006726 001402 BEQ      ILFDON
1883 006730 000167 177304 BEQ      MRLF1
1884 006734                                     JMP      MRLF1
ILFDON:
;FINISHED ALL ILF CODES GET OUT
;START NEXT ILF FUNCTION

```

```

1885 ;*****
1886 ;TEST 40 TEST NO-OP CODES 1 AND 21
1887 ;*****
1888 006734 104400 TST40: SCOPE
1889
1890 ;MODULE TESTED M7759
1891 006736 104414 MR0P: CLRDK ;CLEAR ALL DRIVE REGISTERS
1892 006740 042767 000004 172220 BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
1893 006746 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
1894 006750 104420 MRCK ;CHECK MR REG TO
1895 006752 022701 22701 ;EQUAL 22701
1896 006754 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1897 ;SEND INDEX PULSE
1898 006756 032767 000010 172202 BIT #BIT3,ONCEE ;TESTING CODE I
1899 006764 001031 BNE 3$ ;NO CODE 21
1900 006766 012777 000001 172104 MOV #1,RSRCS1 ;LOAD NO-OP FUNCTION
1901 006774 012767 000001 172212 MOV #1,WORK ;LOAD NO-OP FUNCTION
1902 007002 005777 172106 TST RSRER ;ANY ERRORS
1903 007006 001403 BEQ 1$ ;NO
1904 007010 004767 012252 JSR PC,NOPERR ;TYPE IT
1905 007014 104040 HLT !DS ;TYPE ERROR
1906 007016 022777 010600 172066 1$: CMP #10600,RSRDS ;IS RSDS CORRECT
1907 007024 001403 BEQ 2$ ;YES
1908 007026 004767 012234 JSR PC,NOPERR ;RSDS SHOULD
1909 007032 104040 HLT !DS ;EQUAL 10600
1910 007034 042767 000004 172124 2$: BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
1911
1912 ;TEST NO-OP FUNCTION CODE 21
1913
1914 007042 052767 000010 172116 BIS #BIT3,ONCEE ;TEST TESTING CODE 21 FLAG
1915 007050 012767 000021 172136 3$: MOV #21,WORK ;LOAD CODE 21
1916 007056 012777 000021 172014 MOV #21,RSRCS1 ;LOAD FUNCTION
1917 007064 005777 172024 TST RSRER ;ANY ERRORS?
1918 007070 001403 BEQ 4$ ;NO
1919 007072 004767 012170 JSR PC,NOPERR ;YES, TYPE ERROR
1920 007076 104040 HLT !DS ;ERROR DURING NO-OP FUNCTION
1921 007100 022777 010600 172004 4$: CMP #10600,RSRDS ;IS RSDS CORRECT
1922 007106 001403 BEQ 5$ ;YES
1923 007110 004767 012152 JSR PC,NOPERR ;TYPE ERROR
1924 007114 104040 HLT !DS ;RSDS SHOULD=10600
1925 007116 042767 000014 172042 5$: BIC #14,ONCEE ;CLEAR TEST BITS

```



```

1926
1927
1928
1929 007124 104400
1930
1931
1932 007126 104414
1933 007130 104416
1934 007132 104420
1935 007134 022701
1936 007136 104424
1937 007140 104430
1938
1939 007142 012777 177777 171744
1940 007150 116701 172010
1941 007154 017700 171736
1942 007160 020100
1943 007162 001427
1944 007164 104402 007170
1945 007240 104000
1946 007242 012767 000001 171744 1S:
1947 007250 032767 000010 171710
1948 007256 001004
1949 007260 012777 000001 171612
1950 007266 000406
1951 007270 012767 000021 171716 2S:
1952 007276 012777 000021 171574
1953 007304 017700 171604 3S:
1954 007310 012701 177017
1955 007314 020100
1956 007316 001411
1957 007320 104402 007324
1958 007334 004767 012022
1959 007340 104000
1960 007342 017700 171550 4S:
1961 007346 116701 171612
1962 007352 020100
1963 007354 001411
1964 007356 104402 007362
1965 007372 004767 011764
1966 007376 104000
1967 007400 017700 171506 5S:
1968 007404 012701 150600
1969 007410 020100
1970 007412 001411
1971 007414 104402 007420
1972 007430 004767 011726
1973 007434 104000
1974 007436 032767 000010 171522 6S:
1975 007444 001005
1976 007446 052767 000010 171512
1977 007454 000167 177446
1978 007460 042767 000010 171500 7S:

```

```

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

:MODULE TESTED M7759
MROPER: CLRDK
MRDMD
MRCK
22701
MRINT
MRIND

: CLEAR ALL REGISTERS
: PUT DRIVE INTO MAINT MODE
: CHECK MR REG
: TO EQUAL 22701
: INIT MAINT MODE (CLEAR MRSP)
: SEND INDEX PULSE

: LOAD RSER WITH ERRORS
: GET DRIVE UNDER TEST
: GET RSAS REG
: DID ATA BIT SET CAUSED BY ERROR
: YES
: .ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
: RSAS=BAD GOOD=CORRECT ANS
: SETUP FOR NO-OP CODE 1
: TESTING CODE 21?
: YES
: SEND NO-OP CODE 1
: CHECK FOR ERRORS
: SETUP FOR CODE 21
: SENT NO-OP CODE 21
: GET RSER REG
: GET CORRECT ANS
: DID RSER CHANGE WITH NO-OP
: NO
: .ASCIZ <15><12>"RSER "

: RSER=BAD GOOD=CORRECT ANS
: GET RSAS
: GET CORRECT ANS
: IS RSAS CORRECT
: YES
: .ASCIZ <15><12>"RSAS "
: TYPE ERROR
: RSAS=BAD GOOD=CORRECT ANS
: GET RSDS
: GET CORRECT ANS
: DID RSDS CHANGE
: NO
: .ASCIZ <15><12>"RSDS "
: TYPE ERROR
: RSDS=BAD GOOD=CORRECT ANS
: TESTING CODE 21
: YES, GET OUT
: SET CODE 21 FLAG
: TEST CODE 21
: DONE CLEAR FLAG AND CONT.

```

1979
1980
1981
1982 007466 104400
1983
1984
1985
1986
1987
1988
1989
1990
1991 007470 104414
1992 007472 052767 000040 171466
1993 007500 104416
1994 007502 104420
1995 007504 022701
1996 007506 104424
1997 007510 104430
1998 007512 012777 000003 171370
1999 007520 022777 000002 171400
2000 007526 001403
2001 007530 012777 000041 171352
2002 007536 012777 000031 171334
2003 007544 104426
2004 007546 030400
2005 007550 104000
2006
2007 007552 012767 021506 171422
2008 007560 104422
2009 007562 104426
2010 007564 030400
2011 007566 104000
2012 007570 005367 171406
2013 007574 001371
2014
2015 007576 104422
2016 007600 104426
2017 007602 110600
2018 007604 104000
2019 007606 022777 104230 171264
2020 007614 001401
2021 007616 104140
2022 007620 016777 171336 171270
2023 007626 005777 171264
2024 007632 001401
2025 007634 104140
2026 007636 022777 004230 171234
2027 007644 001401
2028 007646 104140

```

:*****
:TEST 42          BLOCK SEARCH TEST 1
:*****
TST42: SCOPE

:MODULE TESTED: M7759, M7754, M7771, M7770
: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.

MRSRCH: CLDK          ;CLEAR ALL REGISTERS
          BIS          #BITS,ONCEE ;SET CLOCK FLAG
          MRDMD        ;PUT DRIVE INTO MAINTENANCE MOE
          MRCK         ;CHECK MR REG
          22701        ;TO EQUAL 22701
          MRINT        ;INIT MR REG (CLEAR MRSP)
          MRIND        ;CLOCK INDEX PULSE IN RSMR
          MOV          #3,RSDA      ;DO A SEARCH FOR SECTOR 3 OR 41
          CMP          #2,RSDDT     ;INTERLEAVED?
          BEQ          4$          ;NO SECTOR 3
          MOV          #41,RSDA     ;YES SECTOR 41
          MOV          #31,RSCSI    ;LOAD SEARCH COMMAND (M7759)
          DSK         ;CHECK RSDS
          30400        ;TO EQUAL 30400
          HLT         ;PIP SHOULD BE SET AND DRY SHOULD
                   ;BE 0 FOR A DRIVE SEARCH CMD
                   ;STEP THROUGH 3 SECTORS
1$: MOV          #21506,REPT      ;CLOCK MR
   MRCLK         ;RSDS SHOULD NOT
   DSK          ;CHANGE TILL CLOCKING IS COMPLETED
   30400        ;TO REACH SECTOR 3
   HLT         ;KEEP CLOCKING TILL
   DEC          REPT             ;SECTOR 3 HAS BEEN REACHED
   BNE          1$
;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
MRCLK         ;CLOCK MR REG
DSK          ;CHECK FOR "ATA" AND "DRY"
110600       ;TO BE SET IN RSDS FOR
HLT         ;SEARCH FUNCTION SHOULD BE COMPLETED
CMP          #104230,RSCSI      ;SET RSCSI
BEQ          2$          ;SC IN RSCSI SHOULD SET BECAUSE OF
HLT          ;COMPLETED SEARCH FUNCTION
MOV          UNITSV,RSAS        ;CLEAR ATA
TST         RSAS           ;DID ATA CLEAR BY WRITING INTO IT?
BEQ          3$          ;YES
HLT          ;RSAS SHOULD=0
CMP          #4230,RSCSI       ;DID SC CLEAR BY CLEARING
BEQ          +4          ;"ATA" YES
HLT          ;NO

```



```

2029 :*****
2030 :TEST 43          BLOCK SEARCH TEST 2
2031 :*****
2032 007650 104400 TST43: SCOPE
2033
2034 :MODULE TESTED: M7759, M7754, M7771, M7770
2035 :THIS TEST INITIALIZES A BLOCK SEARCH FUNCTION FOR SECTOR 0. WHEN THE DRIVE
2036 :IS CURRENTLY AT THE DESIRED SECTOR. THE BLOCK SEARCH FUNCTION
2037 :SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION
2038 :AND REACHES THE BEGINNING OF THE DESIRED SECTOR.
2039
2040 007652 104414 MRSRC: CLRDK          ;CLEAR ALL REGISTERS
2041 007654 052767 000040 171304 BIS          #BITS,ONCEE ;SET CLOCK FLAG
2042 007662 104416 MRDMD          ;PUT DRIVE INTO MAINTENANCE MOE
2043 007664 104420 MRCK          ;CHECK MR REG
2044 007666 022701 22701          ;TO EQUAL 22701
2045 007670 104424 MRINT          ;INIT MR REG (CLEAR MRSP)
2046 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2047 007672 104430 MRIND
2048 007674 104420 MRCK          ;CHECK MR REG TO EQUAL
2049 007676 022701 22701          ;22701
2050 007700 104000 HLT
2051 ;STEP THRU RESYNC PERIOD
2052 007702 012767 001000 171272 MOV          #512,REPT
2053 007710 052767 000040 171250 BIS          #BITS,ONCEE ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2054 007716 104422 MRRT1: MRCLK        ;CLOCK MR REG
2055 007720 104420 MRCK          ;CHECK FOR
2056 007722 072701 72701          ;CORRECT DATA
2057 007724 104000 HLT          ;MR = BAD GOOD = CORRECT DATA
2058 007726 104422 MRCLK        ;CLOCK MR REG
2059 007730 104420 MRCK          ;CHECK FOR
2060 007732 022701 22701          ;CORRECT DATA
2061 007734 104000 HLT          ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2062 007736 005367 171240 DEC          REPT ;FINISH LOOPING
2063 007742 001365 BNE          MRRT1 ;THROUGH RESYNC PERIOD
2064 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
2065 007744 104422 MRCLK        ;CLOCK MR REG
2066 007746 104420 MRCK          ;MR SHOULD
2067 007750 072301 72301          ;EQUALS 72301
2068 007752 104000 HLT          ;MR=BAD GOOD=CORRECT ANS
2069 007754 104422 MRCLK        ;CLOCK MR REG
2070 007756 104420 MRCK          ;CHECK MR
2071 007760 022301 22301          ;TO EQUAL 22301
2072 007762 104000 HLT          ;MR=BAD GOOD=CORRECT ANS
2073 007764 012767 000100 171210 MOV          #100,REPT ;STEP INTO SECTOR 0
2074 007772 104422 25: MRCLK        ;CLOCK MR REG
2075 007774 005367 171202 DEC          REPT ;DO 100 TIMES
2076 010000 001374 BNE          25 ;DONE YET? NO BR
2077 010002 012777 000031 171070 45: MOV          #31,RSRCS1 ;LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
2078 010010 104426 DSCK          ;CHECK RSDS
2079 010012 030400 30400          ;TO EQUAL 30400
2080 010014 104000 HLT          ;PIP SHOULD BE SET AND DRY SHOULD
2081 ;BE 0 FOR A DRIVE SEARCH CMD
2082 010016 012767 021506 171156 MOV          #21506,REPT ;STEP 3 SECTORS BEYOND SECTOR 0

```

```

2083 010024 104422      1$      MRCLK      ;CLOCK MR
2084 010026 104426      D$CK      ;RSDS SHOULD NOT
2085 010030 030400      30400     ;CHANGE TILL CLOCKING IS COMPLETED
2086 010032 104000      HLT       ;TO REACH SECTOR 3
2087 010034 005367 171142  DEC        REPT  ;KEEP CLOCKING TILL
2088 010040 001371      BNE        1$    ;SECTOR 3 HAS BEEN REACHED
2089      ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2090 010042 104430      MRIND
2091 010044 104420      MRCK
2092 010046 022701      ;CHECK MR REG TO EQUAL
2093 010050 104000      22701
2094      HLT
2095      ;STEP THRU RESYNC PERIOD
2096
2097 010052 012767 001000 171122  MOV        #512,REPT
2098 010060 052767 000040 171100  BIS        #BITS,ONCEE ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2099 010066 104422      MRWR1: MRCLK ;CLOCK MR REG
2100 010070 104420      MRCK      ;CHECK FOR
2101 010072 072701      72701     ;CORRECT DATA
2102 010074 104000      HLT       ;MR = BAD GOOD = CORRECT DATA
2103 010076 104422      MRCLK     ;CLOCK MR REG
2104 010100 104420      MRCK      ;CHECK FOR
2105 010102 022701      22701     ;CORRECT DATA
2106 010104 104000      HLT       ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2107 010106 005367 171070  DEC        REPT  ;FINISH LOOPING
2108 010112 001365      BNE        MRWR1 ;THROUGH RESYNC PERIOD
2109
2110      ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2111      ;SP=0 EQUALS SECTOR PULSE
2112 010114 104422      MRCLK     ;CLOCK MR REG
2113 010116 104420      MRCK      ;MR SHOULD
2114 010120 072301      72301     ;EQUAL 72301
2115 010122 104000      HLT       ;MR=BAD GOOD=CORRECT ANS
2116 010124 104422      MRCLK     ;CLOCK MR REG
2117 010126 104420      MRCK      ;CHECK MR
2118 010130 022301      22301     ;TO EQUAL 22301
2119 010132 104000      HLT       ;MR=BAD GOOD=CORRECT ANS
2120
2121      ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2122 010134 104422      MRCLK     ;CLOCK MR REG
2123 010136 104426      D$CK      ;CHECK FOR "ATA" AND "DRY"
2124 010140 110600      110600    ;TO BE SET IN RSDS FOR
2125 010142 104000      HLT       ;SEARCH FUNCTION SHOULD BE COMPLETED
2126 010144 022777 104230 170726  CMP        #104230,@RSCS1 ;SET RSCS1
2127 010152 001401      BEQ        2$    ;SC IN RSCS1 SHOULD SET BECAUSE OF
2128 010154 104140      HLT       !DS!AS ;COMPLETED SEARCH FUNCTION
2129 010156 016777 171000 170732 2$: MOV        UNITSV,@RSAS ;CLEAR ATA
2130 010164 005777 170726      TST        @RSAS ;DID ATA CLEAR BY WRITING INTO IT?
2131 010170 001401      BEQ        3$    ;YES
2132 010172 104140      HLT       !DS!AS ;RSAS SHOULD=0
2133 010174 022777 004230 170676 3$: CMP        #4230,@RSCS1 ;DID SC CLEAR BY CLEARING
2134 010202 001401      BEQ        +4    ;"ATA" YES
2135 010204 104140      HLT       !DS!AS ;NO

```


F05

MAINDEC-11-DERSD-B
DERSDB.P11 TST44

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 58
DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)

```

2136 ;*****
2137 ;TEST 44 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
2138 ;*****
2139 010206 104400 TST44: SCOPE
2140
2141 ;MODULE TESTED M7759, M7755, M7770
2142 ;RMR ERROR IS CAUSED BY WRITING INTO RSCS1 WHILE DOING A BLOCK SEARCH FUNCTION
2143 ;CHECK RMR DECODER, E12, M7755, IF THIS TEST FAILS
2144
2145 010210 104414 RMR01: CLRDK ;CLEAR ALL DRIVE REGISTERS
2146 010212 042767 000040 170746 BIC #BITS,ONCEE ;CLEAR CLK CNT FLAG
2147 010220 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2148 010222 104420 MRCK ;CHECK MR REG TO
2149 010224 022701 22701 ;EQUAL 22701
2150 010226 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2151 010230 012777 000001 170652 MOV #1,RSDA ;LOAD RSDA
2152 010236 012777 000031 170634 MOV #31,RSCS1 ;LOAD BLOCK SEARCH FUNCTION
2153 010244 104426 DSKC ;CHECK RSDS
2154 010246 030400 30400 ;TO EQUAL 30400
2155 010250 104000 HLT ;DRY IN RSDS SHOULD BE
2156 ;CLEARED FOR DRIVE WAS
2157 ;ISSURED A BLOCK SEARCH FUNCTION
2158 ;RSDS=BAD GOOD=CORRECT ANS
2159 010252 012777 000011 170620 MOV #11,RSCS1 ;LOAD A CLEAR FUNCTION
2160 ;THIS SHOULD CAUSE AN RMR
2161 ;ERROR FOR DRIVE WAS BUSY
2162 ;WHEN CLEAR COMMAND WAS GIVEN
2163 010260 017700 170630 MOV RSER,BAD ;GET RSER REG
2164 010264 012701 000004 MOV #4,GOOD ;GET CORRECT ANS
2165 010270 020100 CMP GOOD,BAD ;DID RMR SET IN RSER?
2166 010272 001410 BEQ 1$ ;YES
2167 010274 104402 021433 TYPE ,TRMR
2168 010300 104402 010304 TYPE ,.+2 ;.ASCIZ "RSCS1"
2169 010312 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
2170 010314 104426 1$: DSKC ;CHECK RSDS TO
2171 010316 150600 150600 ;EQUAL 150600
2172 010320 104000 HLT ;RSDS=BAD GOOD=CORRECT ANS
2173 010322 022777 104230 170550 CMP #104230,RSCS1 ;DID CORRECT BITS SET IN RSCS1
2174 010330 001401 BEQ 2$ ;YES
2175 010332 104040 HLT !DS ;RSCS1 SHOULD=104230
2176 ;RSDS SHOULD=150600
2177 ;RSER SHOULD=4
2178 010334 022777 000001 170546 2$: CMP #1,RSDA ;DID CLR CLEAR RSDA
2179 010342 001401 BEQ 4$ ;NO
2180 010344 104004 HLT !DA ;RSDA SHOULD=1
2181 010346 104414 4$: CLRDK ;CLEAR ALL REGISTERS
2182 010350 005777 170540 TST RSER ;RSER SHOULD CLEAR
2183 010354 001401 BEQ 3$ ;RSER OK
2184 010356 104040 HLT !DS ;RSER SHOULD=0 FOR THE
2185 ;CLEAR BIT WAS LOADED IN RSCS2
2186 010360 022777 004200 170512 3$: CMP #4200,RSCS1 ;RSCS1 SHOULD=4200 FOR THE
2187 010366 001401 BEQ +4 ;CLEAR BIT WAS LOADED IN RSCS2
2188 010370 104040 HLT !DS ;RSCS1 SHOULD=4200

```

G05

MAINDEC-11-DERSD-B
DERSDB.P11 TST45

RS11-R504 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 59
DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)

2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240

010372 104400
010374 104414
010376 104416
010400 104420
010402 022701
010404 104424
010406 012777 000001 170474
010414 012777 000031 170456
010422 104426
010424 030400
010426 104000
010430 005077 170454
010434 017700 170454
010440 012701 000004
010444 020100
010446 001410
010450 104402 021433
010454 104402 010460
010466 104000
010470 104426
010472 150600
010474 104000
010476 022777 104230 170374
010504 001401
010506 104040
010510 022777 000001 170372
010516 001401
010520 104004
010522 104414
010524 005777 170364
010530 001401
010532 104040
010534 022777 004200 170336
010542 001401
010544 104040

```
*****
:TEST 45          DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
*****
TST45: SCOPE

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

RMRC2:  CLRDK          ;CLEAR ALL DRIVE REGISTERS
        MRDMD         ;PUT DRIVE INTO MAINT MODE
        MRCK          ;CHECK MR REG TO
        22701        ;CHECK MR REG TO
        MRINT         ;EQUAL 22701
        MOV           ;INIT MAINT MODE (CLEAR MRSP)
        #1, RSDA     ;LOAD RSDA
        MOV           ;LOAD BLOCK SEARCH FUNCTION
        #31, RSCS1  ;CHECK RSDS
        DSCK         ;TO EQUAL 30400
        30400       ;DRY IN RSDS SHOULD BE
        HLT          ;CLEARED FOR DRIVE WAS
                   ;ISSURED A BLOCK SEARCH FUNCTION
                   ;RSDS=BAD GOOD=CORRECT ANS
                   ;MODIFY RSDA
                   ;THIS SHOULD CAUSE AN RMR
                   ;ERROR FOR DRIVE WAS BUSY
                   ;WHEN COMMAND WAS GIVEN
                   ;GET RSER REG
                   ;GET CORRECT ANS
                   ;DID RMR SET IN RSER?
                   ;YES
                   ;ASCIZ "RSDA"
                   ;RSER=BAD GOOD=CORRECT ANS
                   ;CHECK RSDS TO
                   ;EQUAL 150600
                   ;RSDS=BAD GOOD=CORRECT ANS
                   ;DID CORRECT BITS SET IN RSCS1
                   ;YES
                   ;RSCS1 SHOULD=104230
                   ;RSDS SHOULD=50400
                   ;RSER SHOULD=4
                   ;DID CLR CLEAR RSDA
                   ;NO
                   ;RSDA SHOULD=1
                   ;CLEAR ALL REGISTERS
                   ;RSER SHOULD CLEAR
                   ;RSER OK
                   ;RSER SHOULD=0 FOR THE
                   ;CLEAR BIT WAS LOADED IN RSCS2
                   ;RSCS1 SHOULD=4200 FOR THE
                   ;CLEAR BIT WAS LOADED IN RSCS2
                   ;RSCS1 SHOULD=4200

1S:      DSCK         ;
        150600       ;
        HLT          ;
        CMP           ;#104230, RSCS1
        BEQ          ;2S
        HLT          ;!DS

2S:      CMP           ;#1, RSDA
        BEQ          ;4S
        HLT          ;!DA

4S:      CLDK         ;
        TST          ;RSDA
        BEQ          ;3S
        HLT          ;!DS

3S:      CMP           ;#4200, RSCS1
        BEQ          ;+4
        HLT          ;!DS
```


H05

MAINDEC-11-DERSD-B
DERSDB.P11 TST46

RS11-RSD4 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 60
DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)

```
2241 ;*****
2242 ;TEST 46 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
2243 ;*****
2244 010546 104400 †TST46: SCOPE
2245
2246 ;MODULE TESTED M7759, M7755, M7770
2247 ;RMR ERROR IS CAUSED BY WRITTING INTO RSER WHILE DOING A BLOCK SEARCH FUNCTION
2248 ;CHECK RMR DECODER, E12-M7755, IF THIS TEST FAILS.
2249
2250 010550 104414 RMRC3: CLRDK ;CLEAR ALL DRIVE REGISTERS
2251 010552 042767 000040 170406 BIC #BITS, ONCEE ;CLEAR CLOCK COUNT FLAG
2252 010560 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2253 010562 104420 MRCK ;CHECK MR REG TO
2254 010564 022701 22701 ;EQUAL 22701
2255 010566 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2256 010570 012777 000001 170312 MOV #1, @RSDA ;LOAD RSDA
2257 010576 012777 000031 170274 MOV #31, @RSCS1 ;LOAD BLOCK SEARCH FUNCTION
2258 010604 104426 DSCK ;CHECK RSDS
2259 010606 030400 30400 ;TO EQUAL 30400
2260 010610 104000 HLT ;DRY IN RSDS SHOULD BE
2261 ;CLEARED FOR DRIVE WAS
2262 ;ISSURED A BLOCK SEARCH FUNCTION
2263 ;RSDS=BAD GOOD=CORRECT ANS
2264 010612 012777 177777 170274 MOV #-1, @RSER ;MODIFY RSER
2265 ;THIS SHOULD CAUSE AN RMR
2266 ;ERROR FOR DRIVE WAS BUSY
2267 ;WHEN COMMAND WAS GIVEN
2268 010620 017700 170270 MOV @RSER, BAD ;GET RSER REG
2269 010624 012701 000004 MOV #4, GOOD ;GET CORRECT ANS
2270 010630 020100 CMP GOOD, BAD ;DID RMR SET IN RSER?
2271 010632 001410 BEQ 1$ ;YES
2272 010634 104402 021433 TYPE ,TRMR ;.ASCIZ "RSER"
2273 010640 104402 010644 TYPE ,+2 ;RSER=BAD GOOD=CORRECT ANS
2274 010652 104000 HLT ;CHECK RSDS TO
2275 010654 104426 1$ DSCK ;EQUAL 150600
2276 010656 150600 150600 ;RSDS=BAD GOOD=CORRECT ANS
2277 010660 104000 HLT ;DID CORRECT BITS SET IN RSCS1
2278 010662 022777 104230 170210 CMP #104230, @RSCS1 ;YES
2279 010670 001401 BEQ 4$ ;RSCS1 SHOULD=104230
2280 010672 104040 HLT !DS ;RSDS SHOULD=150600
2281 ;RSER SHOULD=4
2282 ;CLEAR ALL REGISTERS
2283 010674 104414 4$: CLRDK ;CLEAR ALL REGISTERS
2284 010676 005777 170212 TST @RSER ;RSER SHOULD CLEAR
2285 010702 001401 BEQ 3$ ;RSER OK
2286 010704 104040 HLT !DS ;RSER SHOULD=0 FOR THE
2287 ;CLEAR BIT WAS LOADED IN RSCS2
2288 010706 022777 004200 170164 3$: CMP #4200, @RSCS1 ;RSCS1 SHOULD=4200 FOR THE
2289 010714 001401 BEQ +4 ;CLEAR BIT WAS LOADED IN RSCS2
2290 010716 104040 HLT !DS ;RSCS1 SHOULD=4200
```

```

2291 ;*****
2292 ;TEST 47 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
2293 ;*****
2294 010720 104400 TST47: SCOPE
2295
2296 ;MODULE TESTED: M7759, M7755, M7770
2297 ;RMR ERROR SHOULD NOT SET BY WRITTING INTO RSAS WHILE DOING A BLOCK SEARCH FUNCTION
2298 ;IF TEST FAILS, CHECK RMR DECODER E12-M7755.
2299
2300 010722 104414 RMRC4: CLRDK ;CLEAR ALL DRIVE REGISTERS
2301 010724 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2302 010726 104420 MRCK ;CHECK MR REG TO
2303 010730 022701 22701 ;CHECK MR REG TO
2304 010732 104424 MRINT ;EUAL 22701
2305 010734 012777 000001 170146 MOV #1,RSDA ;INIT MAINT MODE (CLEAR MRSP)
2306 010742 012777 000031 170130 MOV #31,RSRCS1 ;LOAD RSDA
2307 010750 104426 DSK 30400 ;LOAD BLOCK SEARCH FUNCTION
2308 010752 030400 HLT ;CHECK RSDS
2309 010754 104000 ;TO EQUAL 30400
2310 ;DRY IN RSDS SHOULD BE
2311 ;CLEARED FOR DRIVE WAS
2312 ;ISSURED A BLOCK SEARCH FUNCTION
2313 010756 005077 170134 CLR RSAS ;RSDS=BAD GOOD=CORRECT ANS
2314 ;WRITE INTO ATTENTION SUMMARY REGISTER.
2315 ;SHOULD BE NO RMR ERROR BECAUSE
2316 010762 017700 170126 MOV RSER,BAD ;WRITING RSAS IS ALLOWED ANYTIME.
2317 010766 012701 000000 MOV #0,GOOD ;GET RSER REG
2318 010772 020100 CMP GOOD,BAD ;GET CORRECT ANS
2319 010774 001435 BEQ 1$ ;DID RMR SET IN RSER?
2320 010776 104402 011002 TYPE ,.+2 ;NO
2321 011066 104000 HLT ;ASCIZ <15><12>"RMR ERROR SHOULD NOT SET WHILE WRITING
2322 011070 104426 1$: DSK 30400 ;RSDS=BAD GOOD=CORRECT ANS
2323 011072 030400 HLT ;CHECK RSDS TO
2324 011074 104000 HLT ;EUAL 30400
2325 011076 022777 004231 167774 CMP #4231,RSRCS1 ;RSDS=BAD GOOD=CORRECT ANS
2326 011104 001401 BEQ 4$ ;DID CORRECT BITS SET IN RSCS1
2327 011106 104040 HLT !DS ;YES
2328 ;RSCS1 SHOULD=4231
2329 ;RSDS SHOULD=30400
2330 ;RSER SHOULD=0
2330 011110 104414 4$: CLRDK ;CLEAR ALL REGISTERS
2331 011112 005777 167776 TST RSER ;RSER SHOULD CLEAR
2332 011116 001401 BEQ 3$ ;RSER OK
2333 011120 104040 HLT !DS ;RSER SHOULD=0 FOR THE
2334 ;CLEAR BIT WAS LOADED IN RSCS2
2335 011122 022777 004200 167750 3$: CMP #4200,RSRCS1 ;RSCS1 SHOULD=4200 FOR THE
2336 011130 001401 BEQ .+4 ;CLEAR BIT WAS LOADED IN RSCS2
2337 011132 104040 HLT !DS ;RSCS1 SHOULD=4200

```


2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464

011446 104400

011450 012767 000002 167466
011456 104414
011460 052767 000040 167500
011466 042767 000600 167472
011474 104430
011476 104420
011500 022701
011502 104424

011504 012702 026572
011510 005022
011512 012722 177777
011516 005003
011520 000261
011522 006103
011524 103402
011526 010322
011530 000774
011532 012703 000156

011536 012704 146314
011542 010422
011544 005303
011546 001375

011550 012777 026572 167330
011556 012777 177600 167320
011564 012777 000061 167306
011572 104446

011574 104220
011576 104450

```
*****
;TEST 51 MAINTENANCE MODE WRITE TEST
*****
TST51: SCOPE

;MODULE TESTED: M7771, M7753, M7751
;THIS IS AN R504 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.

MRWRT: MOV #2,FLAG2 ;SET TEST FLAG
        CLDK ;CLEAR DRIVE REGISTERS
        BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
        BIC #600,ONCEE ;CLEAR FLAG BITS
        MRIND ;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 128 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 (110 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 #110.,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 128 WORDS OF DATA (1 SECTOR) TO SECTOR 0
        MOV #INBUF,ARSBA ;LOAD BUS ADDR REG
        MOV #177600,ARSWC ;LOAD WORD COUNT REG
        MOV #61,ARSCS1 ;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 2 TIMES SP = 1
```

```

2465 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2466 011600 104430 MRIND
2467 011602 104420 MRCK ;CHECK MR REG TO EQUAL
2468 011604 020501 20501 ;20501 FOR A
2469 011606 104000 HLT ;WRITE COMD HAS BEEN ISSUED
2470
2471 ;STEP THRU RESYNC PERIOD
2472
2473 011610 012767 001000 167364 MOV #512.,REPT
2474 011616 052767 000040 167342 BIS #BITS,ONCEE ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
2475 011624 104422 MRWRT1: MRCLK ;CLOCK MR REG
2476 011626 104420 MRCK ;CHECK FOR
2477 011630 070501 70501 ;CORRECT DATA
2478 011632 104000 HLT ;MR = BAD GOOD = CORRECT DATA
2479 011634 104422 MRCLK ;CLOCK MR REG
2480 011636 104420 MRCK ;CHECK FOR
2481 011640 020501 20501 ;CORRECT DATA
2482 011642 104000 HLT ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2483 011644 005367 167332 DEC REPT ;FINISH LOOPING
2484 011650 001365 BNE MRWRT1 ;THROUGH RESYNC PERIOD
2485
2486 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2487 ;SP=0 EQUALS SECTOR PULSE
2488 011652 104422 MRCLK ;CLOCK MR REG
2489 011654 104420 MRCK ;MR SHOULD
2490 011656 070101 70101 ;EQUAL 70101
2491 011660 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2492 011662 104422 MRCLK ;CLOCK MR REG
2493 011664 104420 MRCK ;CHECK MR
2494 011666 020101 20101 ;TO EQUAL 20101
2495 011670 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2496
2497 ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE
2498
2499 011672 012767 000077 167302 MRWRT2: MOV #63.,REPT
2500 011700 104422 MRCLK ;CLOCK MR REG
2501 011702 104420 MRCK ;CHECK MR REG
2502 011704 071501 71501 ;TO EQUAL 71501
2503 011706 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2504 011710 104422 MRCLK ;CLOCK MR REG
2505 011712 104420 MRCK ;CHECK MR REG
2506 011714 021501 21501 ;TO EQUAL 21501
2507 011716 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2508 011720 005367 167256 DEC REPT ;DONE YET
2509 011724 001365 BNE MRWRT2 ;NO LOOP

```



```

2510 ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
2511
2512 011726 104422 MRCLK ;CLOCK MR REG
2513 011730 104420 MRCK ;CHECK MR REG
2514 011732 171501 171501 ;TO EQUAL 171501
2515 011734 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2516 011736 104422 MRCLK ;CLOCK MR REG
2517 011740 104420 MRCK ;MR REG SHOULD
2518 011742 025501 25501 ;EQUAL 25501
2519 011744 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2520 011746 104422 MRCLK
2521 011750 104420 MRCK
2522 011752 175501 175501
2523 011754 104000 HLT
2524 ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
2525 011756 012767 000003 167216 MOV #3,REPT
2526 011764 104422 MRWRT3: MRCLK ;CLOCK MR REG
2527 011766 104420 MRCK ;CHECK MR REG
2528 011770 025501 25501 ;TO EQUAL 25501
2529 011772 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2530 011774 104422 MRCLK ;CLOCK MR REG
2531 011776 104420 MRCK ;CHECK MR REG
2532 012000 175501 175501 ;TO EQUAL 175501
2533 012002 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2534 012004 005367 167172 DEC REPT ;DONE YES?
2535 012010 001365 BNE MRWRT3 ;NO LOOP BACK
2536
2537 ;MOVE DATA WORD INTO RS04 SHIFT REGISTER (M7753)
2538
2539 012012 104422 MRCLK ;CLOCK MR REG
2540 012014 104420 MRCK ;CHECK MR REG
2541 012016 027501 27501 ;TO EQUAL 27501
2542 012020 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2543 012022 104422 MRCLK ;CLOCK MR REG
2544 012024 104420 MRCK ;MR REG SHOULD
2545 012026 123501 123501 ;EQUAL 123501
2546 012030 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2547
2548 ;ENCODE SYNC 1 (M7751)
2549
2550 012032 104422 MRCLK ;CLOCK MR REG
2551 012034 104420 MRCK ;MR REG SHOULD NOW
2552 012036 073501 73501 ;EQUAL 73501
2553 012040 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2554 012042 012705 026572 MOV #INBUF,R5 ;GET STARTING ADDR FOR DATA BUFFER
2555 012046 011504 MOV (R5),R4 ;GET DATA

```

B06

MAINDEC-11-DERSD-B
DERSDB.P11

TST51

RS11-R504 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE MODE WRITE TEST

MACY11 27(732) 04-OCT-76 13:11 PAGE 67

2556	012050	012767	002167	167136		MOV	#1143.,WORK	:DOING A 1 SECTOR TRANSFER 127 WORDS
2557								:18 BITS PER WORD-CLOCK LOOPS
2558								:TAKE CARE OF 2 BITS AT A TIME
2559								:127 TIMES 9 EQUALS 1143 LOOPS
2560								:TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
2561	012056	042767	000200	167102		BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
2562	012064	052767	000100	167074		BIS	#BIT6,ONCEE	:SET 1ST TRANSFER WORD FLAG
2563	012072	104432			1\$:	XBIT		:GET 2 BITS OF DATA
2564	012074	104434				CLKD1		:SEND FIRST CLOCK PULSE
2565								:AND CALCULATE MR REG
2566								:FOR CORRECT DATA (MMDT+MWDB)
2567	012076	104000				HLT		:MR REG NOT CORRECT
2568	012100	104436				CLKD2		:SEND 2ND CLOCK PULSE TO
2569								:COMPLETE TRANSFER OF 2 BITS
2570								:CALCULATE CORRECT ANS FOR
2571								:MR REG (MMDT+MWDB)
2572	012102	104000				HLT		:MR=BAD GOOD=CORRECT ANS
2573	012104	032767	000200	167054		BIT	#BIT7,ONCEE	:ON LAST WORD YET?
2574	012112	001015				BNE	2\$:YES
2575	012114	032767	000400	167044		BIT	#BIT8,ONCEE	:ON CRC WORD YET?
2576	012122	001040				BNE	3\$:YES
2577	012124	005367	167064			DEC	WORK	:DONE WITH 127 WORDS?
2578	012130	001360				BNE	1\$:NO
2579								
2580	012132	052767	000200	167026		BIS	#BIT7,ONCEE	:SET LAST WORD FLAG
2581	012140	012767	000012	167046		MOV	#10.,WORK	:SET UP TO TRANSFER LAST WORD
2582	012146	005367	167042		2\$:	DEC	WORK	:DONE YET?
2583	012152	001347				BNE	1\$:NO
2584								
2585	012154	052767	000400	167004		BIS	#BIT8,ONCEE	:SET TRANSFERRING CRC WORD
2586	012162	042767	000200	166776		BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
2587	012170	004767	011260			JSR	PC,GENCRC	:GENERATE CRC WORD
2588								:AND LEAVE IN "WORK"
2589	012174	012702	026572			MOV	#INBUF,R2	:GO TO END
2590	012200	062702	000400			ADD	#400,R2	:OF DATA BUFFER
2591	012204	016712	167004			MOV	WORK,R2	:LOAD CRC WORD
2592	012210	010205				MOV	R2,R5	:RESET POINTER FOR
2593	012212	162705	000002			SUB	#2,R5	:R5 FOR CRC WD
2594	012216	012767	000012	166770		MOV	#10.,WORK	:SETUP TO XFER CRC
2595	012224	005367	166764		3\$:	DEC	WORK	:DONE YET
2596	012230	001320				BNE	1\$:NO


```

2597 ;EBL SHOULD NOW ASSERT
2598
2599 012232 104422 MRCLK ;CLOCK MR REG TO STOP THROUGH
2600 ;THE R504 DISK SECTOR DEAD BAND AREA
2601 012234 104420 MRCK ;CHECK MR REG
2602 012236 153501 153501 ;TO EQUAL 103501
2603 012240 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2604
2605 ;LOOP 6 TIMES
2606
2607 012242 012767 000006 166732 4$: MOV #6,REPT
2608 012250 104422 4$: MRCLK ;CLOCK MR REG
2609 012252 104420 MRCK ;CHECK MR REG
2610 012254 003501 3501 ;TO EQUAL 53501
2611 012256 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2612 012260 104422 MRCLK ;CLOCK MR REG
2613 012262 104420 MRCK ;CHECK MR REG
2614 012264 153501 153501 ;TO EQUAL 153501
2615 012266 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2616 012270 005367 166706 DEC REPT
2617 012274 001365 BNE 4$ ;DONE LOOPING YET?
2618 ;NO
2619
2620 ;FINISH UP
2621 012276 104422 MRCLK ;CLOCK MR REG
2622 012300 104420 MRCK ;CHECK MR REG
2623 012302 003501 3501 ;TO EQUAL 3501
2624 012304 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
2625 012306 104422 MRCLK ;CLOCK MR REG
2626 012310 104420 MRCK ;CHECK MR REG
2627 012312 151501 151501 ;TO EQUAL 151501
2628 012314 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2629
2630 ;TRANSFER SHOULD NOW BE COMPLETE
2631
2632 012316 104422 MRCLK ;CLOCK MR REG
2633 012320 104420 MRCK ;CHECK MR REG
2634 012322 002701 2701 ;TO EQUAL 2701
2635 012324 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2636
2637 ;NOW TEST CONTROLLER
2638
2639 012326 005777 166546 TST @RSCS1 ;ANY ERRORS?
2640 012332 100001 BPL 5$ ;NO
2641 012334 104014 HLT !DA!WC ;YES
2642 012336 005777 166542 5$: TST @R5WC ;DID WC GO TO 0
2643 012342 001401 BEQ +4 ;YES
2644 012344 104010 HLT !WC ;WC SHOULD BE = TO 0
2645 012346 022777 000001 166534 CMP #1,@R5DA ;DOES R5DA=1
2646 012354 001401 BEQ +4 ;YES
2647 012356 104004 HLT !DA ;R5DA SHOULD=1
2648 012360 032767 000002 166556 BIT #BIT1,FLAG2 ;IN MAINT VERIFY TEST
2649 012366 001002 BNE +6 ;NO
2650 012370 000137 020720 JMP @#MRVR2 ;YES, GO TO VERIFY TEST
    
```

```

2651 :*****
2652 :TEST 52 MAINTENANCE READ TEST
2653 :*****
2654 012374 104400 †TST52: SCOPE
2655
2656 :MODULE TESTED: M7771, M7753, M7751
2657 :THIS IS AN R504 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR
2658 :READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE
2659 :DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS
2660 :NOT TESTED IN MAINTENANCE MODE.)
2661
2662 012376 104414 MRRD: CLRDK ;CLEAR DRIVE REGISTERS
2663 012400 052767 000040 166560 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
2664 012406 042767 147716 166552 BIC #147716,ONCEE ;CLEAR ALL OTHER FLAG BITS
2665 012414 104430 MRIND ;SEND INDEX PULSE TO MR REG
2666 012416 104420 MRCK ;CHECK MR REG
2667 012420 022701 22701 ;TO EQUAL 22701
2668 012422 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2669 ;BY SENDING 2 CLOCK PULSES
2670
2671 012424 005067 166514 CLR FLAG2 ;CLEAR FLAG TEST BITS
2672
2673 :FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
2674 :DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
2675 : A WORD OF ALL 1'S
2676 : FLOATING 1'S PATTERN (16 WORDS)
2677 : A PATTERN OF 146314 (110 WORDS)
2678
2679 012430 012702 026572 MOV #INBUF,R2 ;GET LOCATION OF INBUF
2680 012434 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
2681 012436 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
2682 012442 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
2683 012444 000261 SEC ;A PATTERN OF FLOATING ONES
2684 012446 006103 15: ROL R3 ;GET PATTERN
2685 012450 103402 BCS 25 ;DONE GET OUT
2686 012452 010322 MOV R3,(R2)+ ;FILL BUFFER
2687 012454 000774 BR 15 ;CONT
2688 012456 012703 000156 25: MOV #110,R3 ;FILL REMAINING PORTION OF
2689 012462 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
2690 012466 010422 35: MOV R4,(R2)+ ;LOAD BUFFER
2691 012470 005303 DEC R3 ;DONE YET
2692 012472 001375 BNE 35 ;NO
2693
2694 :NOTE:
2695 :INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ".
2696 :VIA THE MRDB AND MRDT BITS IN RSMR
2697 :OUTBUF IS WHERE THE DATA WORDS FROM THE
:MASSBUS ARE STORED

```


E06

MAINDEC-11-DERSD-B
DERSDB.P11 TST52RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE READ TEST

MACY11 27(732) 04-OCT-76 13:11 PAGE 70

```

2698 ; SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2699
2700 012474 012777 027372 166404 MOV #OUTBUF, @RSBA ; LOAD BUSS ADDR REG
2701 012502 012777 177600 166374 MOV #177600, @RSWC ; LOAD WORD COUNT REG
2702 012510 012777 000071 166362 MOV #71, @RS051 ; LOAD READ COMMAND
2703 012516 012702 000200 MOV #200, R2 ; CLEAR THE OUTBUF TABLE
2704 012522 012703 027372 MOV #OUTBUF, R3 ; SO THAT WHEN THE READ
2705 4$: CLR (R3)+ ; IS FINISHED, WE CAN COMPARE
2706 012530 005023 DEC R2 ; WHAT WE GOT (OUTBUF) WITH WHAT
2707 012532 001375 BNE 4$ ; WE EXPECTED (INBUF).
2708 012534 104446 GETSP ; CLOCK ROUTINE TO GET SECTOR PULSE
2709 ; TO CLEAR OUT COUNTERS AND REGISTERS
2710 ; THAT OTHERWISE COULD NOT BE CLEARED.
2711 012536 104220 HLT !MR ; COULD NOT SET SECTOR PULSE (0)
2712 012540 104450 SPASS ; CLOCK MR REG 2 TIMES SP = 1
2713
2714 ; ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2715 012542 104430 MRIND
2716 012544 104420 MRCK ; CHECK MR REG TO EQUAL
2717 012546 022601 22601 ; 22601 FOR A
2718 012550 104000 HLT ; READ COMD
2719
2720 ; STEP THRU RESYNC PERIOD
2721
2722 012552 012767 001000 166422 MOV #512, REPT
2723 012560 052767 000040 166400 BIS #BITS, ONCEE ; TYPE OUT CLOCK COUNT IF ERRORS OCCUR
2724 012566 104422 MRRD1: MRCLK ; CLOCK MR REG
2725 012570 104420 MRCK ; CHECK FOR
2726 012572 072601 72601 ; CORRECT DATA
2727 012574 104000 HLT ; MR=BAD GOOD=CORRECT DATA
2728 012576 104422 MRCLK ; CLOCK MR REG
2729 012600 104420 MRCK ; CHECK FOR
2730 012602 022601 22601 ; CORRECT DATA
2731 012604 104000 HLT ; ERROR WHILE CLOCKING THROUGH RESYNC
2732 012606 005367 166370 DEC REPT ; FINISH LOOPING
2733 012612 001365 BNE MRRD1 ; THROUGH RESYNC PERIOD
2734
2735 ; ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2736 ; SP=0 EQUALS SECTOR PULSE
2737 012614 104422 MRCLK ; CLOCK MR REG
2738 012616 104420 MRCK ; MR SHOULD
2739 012620 072201 72201 ; EQUAL 72201
2740 012622 104000 HLT ; MR=BAD GOOD=CORRECT ANS
2741 012624 104422 MRCLK ; CLOCK MR REG
2742 012626 104420 MRCK ; CHECK MR
2743 012630 022201 22201 ; TO EQUAL 22201
2744 012632 104000 HLT ; MR=BAD GOOD=CORRECT ANS

```

```

2745 ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
2746
2747 012634 012767 000107 166340 MRRD2: MOV #71.,REPT
2748 012642 104422 MRCLK ;CLOCK MR REG
2749 012644 104420 MRCK ;CHECK MR REG
2750 012646 073601 73601 ;TO EQUAL 73601
2751 012650 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2752 012652 104422 MRCLK ;CLOCK MR REG
2753 012654 104420 MRCK ;CHECK MR REG
2754 012656 023601 23601 ;TO EQUAL 23601
2755 012660 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2756 012662 005367 166314 DEC REPT ;DONE YET
2757 012666 001365 BNE MRRD2 ;NO LOOP
2758 012670 104422 MRCLK ;CLOCK MR REG
2759 012672 104420 MRCK ;CHECK MR REG
2760 012674 073601 73601 ;TO EQUAL 73601
2761 012676 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2762
2763 ;READ SYNC"1"
2764
2765 012700 012777 000055 166216 MOV #55,RSMR
2766 012706 012777 000045 166210 MOV #45,RSMR
2767 012714 104420 MRCK ;CHECK MR REG
2768 012716 023645 23645 ;TO EQUAL
2769 012720 104000 HLT ;CONTENTS OF GOOD
2770 012722 012777 000055 166174 MOV #55,RSMR
2771 012730 012777 000045 166166 MOV #45,RSMR
2772 012736 104420 MRCK
2773 012740 173645 173645
2774 012742 104000 HLT
2775
2776 ;READ DATA
2777 012744 005067 166254 MRRD3: CLR WORK3 ;CLEAR CLOCK COUNT FOR DATA WD
2778 012750 012705 026572 MOV #INBUF,R5 ;GET STARTING ADDRESS FOR DATA BUFFER
2779 012754 162705 000002 SUB #2,R5
2780 012760 012767 000025 166216 MOV #21.,REPT1 ;SETUP COUNTER FOR 1ST SB BIT
2781 012766 012767 002200 166206 MOV #1152.,REPT ;SETUP COUNTER TO TRANSFER
2782 ;128 WORDS-9X128=1152
2783 ;2 CLOCKS PER 2 BITS OF DATA
2784 012774 104444 1S: RBIT ;GET 2 DATA BITS
2785 012776 104440 CLR1 ;CLOCK MR
2786 013000 104000 HLT ;MR REG NOT CORRECT
2787 013002 104442 CLR2 ;CLOCK MR REG
2788 013004 104000 HLT ;MR REG NOT CORRECT
2789 013006 005367 166170 DEC REPT ;DONE WITH DATA BUFFER YET?
2790 013012 001370 BNE 1S ;NO
    
```



```

2791 013014 032767 000400 166144 2$: BIT #BIT8,ONCEE ;DID WE ALREADY DO CRC?
2792 013022 001030 BNE 3$ ;YES
2793 013024 052767 000400 166134 BIS #BIT8,ONCEE ;NO SET CRC FLAG
2794 013032 016767 166146 166132 MOV REPT1,SAVEE ;SAVE REPT1
2795 013040 004767 010410 JSR PC,GENCRC ;GENERATE CRC WORD
2796 ;AND LEAVE IN LOC "WORK"
2797 013044 012702 026572 MOV #INBUF,R2
2798 013050 016767 166116 166126 MOV SAVEE,REPT1 ;RESTORE REPT1
2799 013056 062702 000400 ADD #400,R2 ;STORE CRC WORD AT END OF
2800 013062 016712 166126 MOV WORK,R2 ;INBUF TABLE
2801 013066 010205 MOV R2,R5
2802 013070 162705 000002 SUB #2,R5
2803 013074 012767 000011 166100 MOV #9.,REPT ;SETUP TO TRANSFER 1 WD
2804 013102 000734 BR 1$ ;TRANSFER CRC WD
2805 013104 104422 3$: MRCLK ;CLOCK MR REG
2806 013106 104420 MRCK ;CHECK MR REG
2807 013110 003601 3601 ;TO EQUAL
2808 013112 104000 HLT ;3601
2809 013114 104422 MRCLK ;CLOCK MR REG
2810 013116 104420 MRCK ;CHECK MR
2811 013120 153601 153601 ;TO EQUAL
2812 013122 104000 HLT ;153601
2813 013124 104422 MRCLK ;CLOCK MR REG
2814 013126 104420 MRCK ;CHECK MR
2815 013130 007601 7601 ;TO EQUAL
2816 013132 104000 HLT ;7601
2817 013134 104422 MRCLK ;CLOCK MR REG
2818 013136 104420 MRCK ;CHECK MR
2819 013140 153601 153601 ;TO EQUAL
2820 013142 104000 HLT ;153601
2821
2822 ;PERFORM 3 DOUBLE MAINTENANCE CLOCK OPERATIONS
2823 ;STEP INTO END OF SECTOR DEAD BAND
2824 ;EBL IS NOW ASSERTED
2825
2826 013144 012767 000010 166030 MRD4: MOV #8.,REPT
2827 013152 104422 1$: MRCLK ;CLOCK MR REG
2828 013154 104420 MRCK ;CHECK MR REG
2829 013156 003601 3601 ;TO EQUAL
2830 013160 104000 HLT ;3601
2831 013162 104422 MRCLK ;CLOCK MR REG
2832 013164 104420 MRCK ;CHECK MR
2833 013166 153601 153601 ;REG TO
2834 013170 104000 HLT ;EQUAL 153601
2835 013172 005367 166004 DEC REPT ;DONE YET?
2836 013176 001365 BNE 1$ ;NO
2837
2838 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
2839 ;SHOULD GET STROBE BUFFER
2840
2841 013200 104422 MRCLK ;CLOCK MR REG
2842 013202 104420 MRCK ;CHECK MR
2843 013204 007601 7601 ;REG TO
2844 013206 104000 HLT ;EQUAL 7601

```

H06

MAINDEC-11-DERSD-B
DERSDB.P11 TST52

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE READ TEST

MACY11 27(732) 04-OCT-76 13:11 PAGE 73

```

2845          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
2846          ;SHOULD COMPLETE TRANSFER.
2847
2848 013210 104422 MRD5: MRCLK          ;CLOCK MR REG
2849 013212 022777 004270 165660 CMP      #4270,RSRCS1 ;ANY ERRORS?
2850 013220 001401          BEQ      1$          ;NO
2851 013222 104054          HLT      !DA!DS!WC
2852 013224 005777 165654 1$:      TST      RSWC          ;DID WC GO TO 0
2853 013230 001401          BEQ      .+4          ;YES
2854 013232 104010          HLT      !WC          ;WC REG SHOULD=0
2855 013234 022777 000001 165646 CMP      #1,RSRDA ;DOES RSDA=1
2856 013242 001401          BEQ      .+4          ;YES
2857 013244 104004          HLT      !DA          ;RSDA SHOULD=1
2858
2859          ;COMPARE DATA READ WITH INPUT BUFFER
2860          ;WILL ONLY TYPEOUT 10 ERRORS --- BUT IF SW12 IS SET
2861          ;IT WILL TYPE OUT ALL ERRORS
2862
2863 013246 012700 026572 MRD6: MOV      #INBUF,BAD ;GET STARTING LOC OF EXPECTED DATA
2864 013252 012701 027372 MOV      #OUTBUF,GOOD ;GET STARTING LOC OF DATA "READ" FROM DISK
2865 013256 012767 000012 165716 MOV      #12,REPT ;SET UP ERROR COUNTER
2866 013264 012705 000201          MOV      #201,R5 ;COMPARE 1 SECTOR
2867 013270 005305 3$:      DEC      R5 ;DONE WITH SECTOR
2868 013272 001433          BEQ      2$          ;YES GET OUT
2869 013274 022021          CMP      (BAD)+,(GOOD)+ ;IS DATA CORRECT?
2870 013276 001774          BEQ      3$          ;YES
2871 013300 032777 010000 164262 BIT      #BIT12,RSR ;TYPE ALL ERRORS?
2872 013306 001003 1$:      BNE      1$          ;YES
2873 013310 005367 165666          DEC      REPT ;TYPED OUT 10 ERRORS YET?
2874 013314 001422          BEQ      2$          ;YES GET OUT
2875 013316 024041 1$:      CMP      -(BAD),-(GOOD) ;GET ERROR
2876 013320 104000          HLT          ;TYPE OUT ERROR
2877 013322 010067 165666          MOV      BAD,WORK
2878 013326 104402 013332          TYPE      .+2 ;ASCIZ "BAD ADDRESS="
2879 013350 016746 165640          MOV      WORK,-(6) ;PUT WORK ON STACK
2880 013354 104406          TYPES ;TYPE STACK IN OCTAL - SUPRESS
2881 013356 022021          CMP      (BAD)+,(GOOD)+
2882 013360 000743          BR      3$
2883 013362 2$:          ;DONE

```



```

2884 ;*****
2885 ;TEST 53 MAINTENANCE MODE DATA WRITE CHECK TEST
2886 ;*****
2887 013362 104400 †TST53: SCOPE
2888
2889 ;MODULE TESTED: M7771, M7753, M7751
2890 ;A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION.
2891 ;WITHIN THE RS04, A WRITE CHECK FUNCTION IS IDENTICAL TO A
2892 ;READ FUNCTION.
2893
2894 013364 104414 MRWCK: CLRDK ;CLEAR DRIVE REGISTERS
2895 013366 052767 000040 165572 BIS #BITS ONCEE ;SET TYPE CLOCK COUNT FLAG
2896 013374 042767 147716 165564 BIC #147716,ONCEE ;CLEAR ALL OTHER FLAG BITS
2897 013402 104430 MRIND ;SEND INDEX PULSE TO MR REG
2898 013404 104420 MRCK ;CHECK MR REG
2899 013406 022701 22701 ;TO EQUAL 22701
2900 013410 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2901 ;BY SENDING 2 CLOCK PULSES
2902
2903 013412 012767 000004 165524 MOV #4,FLAG2 ;SET WC FLAG FOR CLKR1 ROUTINE
2904
2905 ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
2906 ;DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
2907 ; :A WORD OF ALL 1'S
2908 ; :FLOATING 1'S PATTERN (16 WORDS)
2909 ; :A PATTERN OF 146314 (110 WORDS)
2910 ;
2911 013420 012702 026572 MOV #INBUF,R2 ;GET LOCATION OF INBUF
2912 013424 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
2913 013426 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
2914 013432 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
2915 013434 000261 SEC ;A PATTERN OF FLOATING ONES
2916 013436 006103 1S: ROL R3 ;GET PATTERN
2917 013440 103402 BCS 2S ;DONE GET OUT
2918 013442 010322 MOV R3,(R2)+ ;FILL BUFFER
2919 013444 000774 BR 1S ;CONT
2920 013446 012703 000156 2S: MOV #110,R3 ;FILL REMAINING PORTION OF
2921 013452 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
2922 013456 010422 3S: MOV R4,(R2)+ ;LOAD BUFFER
2923 013460 005303 DEC R3 ;DONE YET
2924 013462 001375 BNE 3S ;NO
2925
2926 ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2927
2928 013464 012777 026572 165414 MOV #INBUF,ARSBA ;LOAD BUS ADDR REG
2929 013472 012777 177600 165404 MOV #177600,ARSWC ;LOAD WORD COUNT REG
2930 013500 012777 000051 165372 MOV #51,ARSCS1 ;LOAD WRITE CHECK COMMAND
2931 013506 104446 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
2932 ;TO CLEAR OUT COUNTERS AND REGISTERS
2933 ;THAT OTHERWISE COULD NOT BE CLEARED.
2934 013510 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
2935 013512 104450 SPASS ;CLOCK MR SECTOR PULSE = 1

```



```

2985                                     ;READ SYNC"1"
2986
2987 013652 012777 000055 165244      MOV      #55,ARSMR
2988 013660 012777 000045 165236      MOV      #45,ARSMR
2989 013666 104420                    MRCK
2990 013670 023745                    23745      ;CHECK MR FOR
2991 013672 104000                    HLT        ;FOR CORRECT
2992 013674 012777 000055 165222      MOV      #55,ARSMR      ;ANS IS IN GOOD
2993 013702 012777 000045 165214      MOV      #45,ARSMR
2994 013710 104420                    MRCK
2995 013712 173745                    173745
2996 013714 104000                    HLT
2997
2998                                     ;READ DATA
2999 013716 005067 165302      MRWCK3: CLR      WORK3      ;CLEAR CLOCK COUNT FOR DATA WD
3000 013722 012705 026572      MOV      #INBUF,R5      ;GET STARTING ADDRESS FOR DATA BUFFER
3001 013726 162705 000002      SUB
3002 013732 012767 000025 165244      MOV      #2,R5
3003 013740 012767 002200 165234      MOV      #21,REPT1      ;SETUP COUNTER FOR 1ST SB BIT
3004                                     ;SETUP COUNTER TO TRANSFER
3005                                     ;128 WORDS-9X128=1152
3006 013746 104444                    1$:      RBIT      ;2 CLOCKS PER 2 BITS OF DATA
3007 013750 104440                    CLKR1     ;GET 2 DATA BITS
3008 013752 104000                    HLT      ;CLOCK MR
3009 013754 104442                    CLKR2     ;MR REG NOT CORRECT
3010 013756 104000                    HLT      ;CLOCK MR REG
3011 013760 005367 165216      DEC      #REPT1          ;MR REG NOT CORRECT
3012 013764 001370                    BNE      ;DONE WITH DATA BUFFER YET?
3013 013766 032767 000400 165172 2$: BIT      #BIT8,ONCEE ;NO
3014 013774 001030                    BNE      3$              ;DID WE ALREADY DO CRC?
3015 013776 052767 000400 165162 3$: BIS      #BIT8,ONCEE ;YES
3016 014004 016767 165174 165160      MOV      #NO SET CRC FLAG
3017 014012 004767 007436      JSR      REPT1,SAVEE ;SAVE REPT1
3018                                     ;GENERATE CRC WORD
3019 014016 012702 026572      MOV      #INBUF,R2      ;AND LEAVE IN LOC "WORK"
3020 014022 016767 165144 165154      MOV      SAVEE,REPT1 ;RESTORE REPT1
3021 014030 062702 000400      ADD      #400,R2        ;STORE CRC WORD TO BE READ
3022 014034 016712 165154      MOV      WORK,AR2      ;AT END OF INBUF TABLE
3023 014040 010205                    MOV      R2,R5
3024 014042 162705 000002      SUB      #2,R5
3025 014046 012767 000011 165126      MOV      #9,REPT        ;SETUP TO TRANSFER 1 WD
3026 014054 000734                    BR       1$              ;TRANSFER CRC WD
    
```

```

3027 014056 104422          3$: MRCLK          ;CLOCK MR REG
3028 014060 104420          MRCK           ;CHECK MR REG
3029 014062 003701          3701          ;TO EQUAL
3030 014064 104000          HLT           ;3701
3031 014066 104422          MRCLK         ;CLOCK MR REG
3032 014070 104420          MRCK         ;CHECK MR
3033 014072 153701          153701       ;TO EQUAL
3034 014074 104000          HLT           ;153701
3035 014076 104422          MRCLK         ;CLOCK MR REG
3036 014100 104420          MRCK         ;CHECK MR
3037 014102 007701          7701         ;TO EQUAL
3038 014104 104000          HLT           ;7701
3039 014106 104422          MRCLK         ;CLOCK MR REG
3040 014110 104420          MRCK         ;CHECK MR
3041 014112 153701          153701       ;TO EQUAL
3042 014114 104000          HLT           ;153701
3043
3044          ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3045          ;STEP INTO END OF SECTOR DEAD BAND
3046          ;EBL IS NOW ASSERTED
3047
3048 014116 012767 000010 165056 MRWCK4: MOV      #8.,REPT
3049 014124 104422          1$: MRCLK          ;CLOCK MR REG
3050 014126 104420          MRCK         ;CHECK MR REG
3051 014130 003701          3701         ;TO EQUAL
3052 014132 104000          HLT           ;3601
3053 014134 104422          MRCLK         ;CLOCK MR REG
3054 014136 104420          MRCK         ;CHECK MR
3055 014140 153701          153701       ;REG TO
3056 014142 104000          HLT           ;EQUAL 153601
3057 014144 005367 165032 DEC      REPT
3058 014150 001365          BNE      1$   ;DONE YET?
3059          ;NO
3060          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3061          ;SHOULD GET STROBE BUFFER
3062
3063 014152 104422          MRCLK         ;CLOCK MR REG
3064 014154 104420          MRCK         ;CHECK MR
3065 014156 007701          7701         ;REG TO
3066 014160 104000          HLT           ;EQUAL 7601
3067
3068          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3069          ;SHOULD COMPLETE TRANSFER.
3070
3071 014162 104422          MRWCK5: MRCLK        ;CLOCK MR REG
3072 014164 022777 004250 164706 CMP      #4250,RS0CS1 ;ANY ERRORS?
3073 014172 001401          BEQ      1$     ;NO
3074 014174 104054          HLT      !DA!DS!WC
3075 014176 005777 164702          1$: TST      RS0WC   ;DID WC GO TO 0
3076 014202 001401          BEQ      +4     ;YES
3077 014204 104010          HLT      !WC    ;WC REG SHOULD=0
3078 014206 022777 000001 164674 CMP      #1,RS0DA  ;DOES RSDA=1
3079 014214 001401          BEQ      +4     ;YES
3080 014216 104004          HLT      !DA    ;RSDA SHOULD=1

```


3081
3082
3083
3084 014220 104400
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095 014222 012767 000040 164714
3096 014230 104414
3097 014232 052767 000040 164726
3098 014240 042767 147716 164720
3099 014246 104430
3100 014250 104420
3101 014252 022701
3102 014254 104424
3103
3104 014256 032767 000020 164660
3105 014264 001023
3106 014266 012767 000001 164676
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121 014274 012702 026572
3122 014300 012703 000016
3123 014304 005022
3124 014306 005303
3125 014310 001375
3126 014312 012722 000012
3127 014316 012722 020000
3128 014322 012703 000200
3129 014326 005022
3130 014330 005303
3131 014332 001375

:TEST 54 MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)

TST54: SCOPE

:MODULES TESTED: M7753)
:THE RS04 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.

MRCRC: MOV #40,FLAG2 ;CLEAR TST FLAG
CLRDK ;CLEAR DRIVE REGISTERS
BIS #BITS ONCEE ;TYPE CLOCK COUNT IF ERROR OCCURS
BIC #147716,ONCEE ;CLEAR ALL OTHER FLAG BITS
MRIND ;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
BIT #BIT4,FLAG2 ;FIRST TIME THROUGH
BNE 3\$;NO
MOV #1,SAVEE ;LOAD 1ST CRC WORD

:FILL MEMORY DATA BUFFER (INBUF) WITH 1 SECTOR. CREATE BUFFER
:WITH 144 WORDS OF 16 BITS WHICH EQUALS THE NO. OF BITS IN 128 18 BITS WORDS
:DATA BUFFER CONTAINS 14 WORDS OF ZEROS
: A WORD OF 12
: A WORD OF 20000
: 128 WORDS OF ZEROS

:NOTE:
:IN THIS TEST, ALL 18 BITS OF THE RS04 DATA
:WORD MUST BE MANIPULATED. HENCE, A TABLE
:CONTAINING 2304 BITS (128 X 18) IS REQUIRED
:INSTEAD OF A TABLE CONTAINING 128 WORDS.

1\$: MOV #INBUF,R2 ;GET LOCATION OF INBUF
MOV #14,R3 ;SETUP COUNTER
CLR (R2)+ ;TO CLEAR THE
DEC R3 ;FIRST 14
BNE 1\$;WORDS
MOV #12,(R2)+ ;LOAD A 12
MOV #20000,(R2)+ ;LOAD A 20000
2\$: MOV #128,R3 ;SETUP COUNTER
CLR (R2)+ ;TO CLEAR THE
DEC R3 ;REMAINING WORDS
BNE 2\$;FOR THAT SECTOR

```

3132                                     ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3133
3134 014334 012777 027372 164544 3$: MOV #OUTBUF,ARSBA ;LOAD BUS ADDR REG
3135 014342 012777 177600 164534 MOV #177600,ARSWC ;LOAD WORD COUNT REG
3136 014350 012777 000071 164522 MOV #71,ARSCS1 ;LOAD READ COMMAND
3137 014356 012702 000200 MOV #200,R2
3138 014362 012703 027372 MOV #OUTBUF,R3
3139 014366 052767 000020 164550 BIS #BIT4,FLAG2 ;SET 1ST TIME THROUGH FLAG
3140 014374 005023 4$: CLR (R3)+
3141 014376 005302 DEC R2
3142 014400 001375 BNE 4$
3143 014402 104446 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3144 ;TO CLEAR OUT COUNTERS AND REGISTERS
3145 ;THAT OTHERWISE COULD NOT BE CLEARED.
3146 014404 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3147 014406 104450 SPASS ;CLOCK MR REG SP = 1
3148
3149 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3150 014410 104430 MRIND
3151 014412 104420 MRCK ;CHECK MR REG TO EQUAL
3152 014414 022601 22601 FOR A
3153 014416 104000 HLT ;READ COMD
3154
3155 ;STEP THRU RESYNC PERIOD
3156
3157 014420 012767 001000 164554 MOV #512,REPT
3158 014426 052767 000040 164532 BIS #BIT5,ONCEE ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3159 014434 104422 MRCRC1: MRCLK ;CLOCK MR REG
3160 014436 104420 MRCK ;CHECK FOR
3161 014440 072601 72601 ;CORRECT DATA
3162 014442 104000 HLT ;MR=BAD GOOD=CORRECT DATA
3163 014444 104422 MRCLK ;CLOCK MR REG
3164 014446 104420 MRCK ;CHECK FOR
3165 014450 022601 22601 ;CORRECT DATA
3166 014452 104000 HLT ;ERROR WHILE CLOCKING THROUGH RESYNC
3167 014454 005367 164522 DEC REPT ;FINISH LOOPING
3168 014460 001365 BNE MRCRC1 ;THROUGH RESYNC PERIOD
3169
3170 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3171 ;SP=0 EQUALS SECTOR PULSE
3172 014462 104422 MRCLK ;CLOCK MR REG
3173 014464 104420 MRCK ;MR SHOULD
3174 014466 072201 72201 ;EQUAL 72201
3175 014470 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3176 014472 104422 MRCLK ;CLOCK MR REG
3177 014474 104420 MRCK ;CHECK MR
3178 014476 022201 22201 ;TO EQUAL 22201
3179 014500 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```



```

3180                                     ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3181
3182 014502 012767 000107 164472      MRCRC2: MOV      #71.,REPT
3183 014510 104422                    MRCLK
3184 014512 104420                    MRCK
3185 014514 073601                    73601
3186 014516 104000                    HLT
3187 014520 104422                    MRCLK
3188 014522 104420                    MRCK
3189 014524 023601                    23601
3190 014526 104000                    HLT
3191 014530 005367 164446          DEC      REPT
3192 014534 001365                    BNE     MRCRC2
3193 014536 104422                    MRCLK
3194 014540 104420                    MRCK
3195 014542 073601                    73601
3196 014544 104000                    HLT
3197
3198                                     ;READ SYNC"1"
3199
3200 014546 012777 000055 164350      MOV      #55,DRSMR
3201 014554 012777 000045 164342      MOV      #45,DRSMR
3202 014562 104420                    MRCK
3203 014564 023645                    23645
3204 014566 104000                    HLT
3205 014570 012777 000055 164326      MOV      #55,DRSMR
3206 014576 012777 000045 164320      MOV      #45,DRSMR
3207 014604 104420                    MRCK
3208 014606 173645                    173645
3209 014610 104000                    HLT
3210
3211                                     ;READ DATA
3212 014612 005067 164406          MRCRC3: CLR      WORK3
3213 014616 012705 026572          MOV      #INBUF,R5
3214 014622 162705 000002          SUB      #2,R5
3215 014626 012767 000025 164350      MOV      #21.,REPT1
3216 014634 012767 002200 164340      MOV      #1152.,REPT
3217
3218                                     ;
3219 014642 104444                    RBIT
3220 014644 104440                    CLR1
3221 014646 104000                    HLT
3222 014650 104442                    CLR2
3223 014652 104000                    HLT
3224 014654 005367 164322          DEC      REPT
3225 014660 001370                    BNE     IS

```

```

;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 73601
;MR=BAD GOOD=CORRECT ANS
;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 23601
;MR=BAD GOOD=CORRECT ANS
;DONE YET
;NO LOOP
;CLOCK MR REG
;CHECK MR REG
;TO EQUAL 73601
;MR=BAD GOOD=CORRECT ANS

;CHECK MR REG
;FOR CORRECT
;ANS IS IN GOOD

;CLEAR CLOCK COUNT FOR DATA WD
;GET STARTING ADDRESS FOR DATA BUFFER

;SETUP COUNTER FOR 1ST SB BIT
;SETUP COUNTER TO TRANSFER
;128 WORDS-9X128=1152
;2 CLOCKS PER 2 BITS OF DATA
;GET 2 DATA BITS
;CLOCK MR
;MR REG NOT CORRECT
;CLOCK MR REG
;MR REG NOT CORRECT
;DONE WITH DATA BUFFER YET?
;NO

```

3226	014662	032767	000400	164276	2S:	BIT	#BIT8, ONCEE	: DID WE ALREADY DO CRC?
3227	014670	001020				BNE	3S	: YES
3228	014672	052767	000400	164266		BIS	#BIT8, ONCEE	: NO SET CRC FLAG
3229	014700	012702	026572			MOV	#INBUF, R2	: MOVE CRC
3230	014704	062702	000440			ADD	#440, R2	: WORD TO END OF
3231	014710	016712	164256		4S:	MOV	SAVEE, R2	: INBUF TABLE
3232	014714	010205			5S:	MOV	R2, R5	: GET CRC WORD
3233	014716	162705	000002			SUB	R2, R5	
3234	014722	012767	000011	164252		MOV	R9, REPT	: SETUP TO TRANSFER 1 WD
3235	014730	000744				BR	1S	: TRANSFER CRC WD
3236	014732	104422			3S:	MRCLK		: CLOCK MR REG
3237	014734	104420				MRCK		: CHECK MR REG
3238	014736	003601				3601		: TO EQUAL
3239	014740	104000				HLT		: 3601
3240	014742	104422				MRCLK		: CLOCK MR REG
3241	014744	104420				MRCK		: CHECK MR
3242	014746	153601				153601		: TO EQUAL
3243	014750	104000				HLT		: 153601
3244	014752	104422				MRCLK		: CLOCK MR REG
3245	014754	104420				MRCK		: CHECK MR
3246	014756	007601				7601		: TO EQUAL
3247	014760	104000				HLT		: 7601
3248	014762	104422				MRCLK		: CLOCK MR REG
3249	014764	104420				MRCK		: CHECK MR
3250	014766	153601				153601		: TO EQUAL
3251	014770	104000				HLT		: 153601


```

3252          :PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3253          :STEP INTO END OF SECTOR DEAD BAND
3254          :EBL IS NOW ASSERTED
3255
3256 014772 012767 000010 164202 MRCRC4: MOV      #8.,REPT
3257
3258 015000 104422          1S:   MRCLK          :CLOCK MR REG
3259 015002 104420          MRCK           :CHECK MR REG
3260 015004 003601          3601          :TO EQUAL
3261 015006 104000          HLT           :3601
3262 015010 104422          MRCLK          :CLOCK MR REG
3263 015012 104420          MRCK           :CHECK MR
3264 015014 153601          153601         :REG TO
3265 015016 104000          HLT           :EQUAL 153601
3266 015020 005367 164156  DEC      REPT      :DONE YET?
3267 015024 001365          BNE      1S      :NO
3268
3269          :PERFORM ONE MAINTENANCE CLOCK OPERATION
3270          :SHOULD GET STROBE BUFFER
3271
3272 015026 104422          MRCLK          :CLOCK MR REG
3273 015030 104420          MRCK           :CHECK MR
3274 015032 007601          7601          :REG TO
3275 015034 104000          HLT           :EQUAL 7601
3276
3277          :PERFORM ONE MAINTENANCE CLOCK OPERATION
3278          :SHOULD COMPLETE TRANSFER.
3279
3280 015036 104422          MRCRC5: MRCLK          :CLOCK MR REG
3281 015040 022777 004270 164032  CMP      #4270,RSRCS1 :ANY ERRORS?
3282 015046 001401          1S          :NO
3283 015050 104054          HLT          :DA!DS!WC
3284 015052 005777 164026  1S:   TST      RSWC      :DID WC GO TO 0
3285 015056 001401          BEQ      +4      :YES
3286 015060 104010          HLT          :WC REG SHOULD=0
3287 015062 006167 164104  ROL      SAVEE     :GET NEXT CRC WORD
3288 015066 103404          BCS      2S      :DONE - BRANCH
3289 015070 004767 010472  JSR      PC,MDATA  :SHIFT DATA PATTERN
3290 015074 000167 177130  JMP      MRCRC    :RESTART TEST WITH NEW DATA PATTERN
3291 015100          2S:

```

E07

MAINDEC-11-DERSD-B
DERSDB.P11 TST55

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 83
MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

```

3292
3293
3294
3295 015100 104400
3296
3297
3298
3299
3300
3301
3302 015102 012767 000040 164034
3303 015110 104414
3304 015112 052767 000040 164046
3305 015120 042767 147716 164040
3306 015126 104430
3307 015130 104420
3308 015132 022701
3309 015134 104424
3310
3311 015136 032767 000020 164000
3312 015144 001023
3313 015146 012767 000001 164016
3314
3315
3316
3317
3318
3319
3320 015154 012702 026572
3321 015160 012703 000017
3322 015164 005022
3323 015166 005303
3324 015170 001375
3325 015172 012722 000001
3326 015176 012722 042000
3327 015202 012703 000177
3328 015206 005022
3329 015210 005303
3330 015212 001375
3331
3332 015214 012777 027372 163664
3333 015222 012777 177600 163654
3334 015230 012777 000071 163642
3335 015236 012702 000200
3336 015242 012703 027372
3337 015246 052767 000020 163670
3338 015254 005023
3339 015256 005302
3340 015260 001375
3341 015262 104446
3342
3343
3344 015264 104220
3345 015266 104450

```

```

;*****
;TEST 55 MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
;*****
TST55: SCOPE
;MODULE TESTED M7753
;THIS TEST IS SIMILAR TO CRC TEST 1 EXECPT 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.
MRDCK: MOV #40,FLAG2 ;CLEAR TST FLAG
;CLEAR DRIVE REGISTERS
CLDRK ;TYPE CLOCK COUNT IF ERROR OCCURS
BIS #BITS,ONCEE ;CLEAR ALL OTHER FLAG BITS
BIC #147716,ONCEE ;SEND INCRX PULSE TO MR REG
MRIND ;CHECK MR REG
MRCK ;TO EQUAL 22701
22701 ;INIT MAINT MODE (CLEAR MRSP)
MRINT ;BY SENDING 2 CLOCK PULSES
;FIRST TIME THROUGH
BIT #BIT4,FLAG2 ;NO
BNE 3$ ;LOAD 1ST CRC WORD
MOV #1,SAVEE ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR) CREATE BUFFER
;WITH 144 WORDS OF 16 BITS WHICH = THE NO. OF BITS IN 128 18 BIT WORDS
;DATA BUFFER CONTAINS 15 WORDS OF ZEROS
; A WORD OF 1
; A WORD OF 42000
; 127 WORDS OF ZEROS
MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
MOV #15,R3 ;SETUP COUNTER
1$: CLR (R2)+ ;TO CLEAR THE
DEC R3 ;FIRST 15
BNE 1$ ;WORDS
MOV #1,(R2)+ ;LOAD A 1
MOV #42000,(R2)+ ;LOAD A 42000
MOV #127,R3 ;SETUP COUNTER
2$: CLR (R2)+ ;TO CLEAR THE
DEC R3 ;REMAINING WORDS
BNE 2$ ;FOR THAT SECTOR
;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3$: MOV #OUTBUF,ARSBA ;LOAD BUS ADDR REG
MOV #177600,ARSWC ;LOAD WORD COUNT REG
MOV #71,ARSCS1 ;LOAD READ COMMAND
MOV #200,R2
MOV #OUTBUF,R3
BIS #BIT4,FLAG2 ;SET 1ST TIME THROUGH FLAG
4$: CLR (R3)+
DEC R2
BNE 4$
GETSP
;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
HLT !MR
SPASS

```



```

3377                                     ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3378
3379 015362 012767 000107 163612 MRDCK2: MOV #71.,REPT
3380 015370 104422 MRCLK ;CLOCK MR REG
3381 015372 104420 MRCK ;CHECK MR REG
3382 015374 073601 73601 ;TO EQUAL 73601
3383 015376 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3384 015400 104422 MRCLK ;CLOCK MR REG
3385 015402 104420 MRCK ;CHECK MR REG
3386 015404 023601 23601 ;TO EQUAL 23601
3387 015406 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3388 015410 005367 163566 DEC REPT ;DONE YET
3389 015414 001365 BNE MRDCK2 ;NO LOOP
3390 015416 104422 MRCLK ;CLOCK MR REG
3391 015420 104420 MRCK ;CHECK MR REG
3392 015422 073601 73601 ;TO EQUAL 73601
3393 015424 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3394
3395                                     ;READ SYNC"1"
3396
3397 015426 012777 000055 163470 MOV #55,RSMR
3398 015434 012777 000045 163462 MOV #45,RSMR
3399 015442 104420 MRCK ;CHECK MR REG
3400 015444 023645 23645 ;TO EQUAL
3401 015446 104000 HLT ;CORRECT ANS IN GOOD
3402 015450 012777 000055 163446 MOV #55,RSMR
3403 015456 012777 000045 163440 MOV #45,RSMR
3404 015464 104420 MRCK
3405 015466 173645 173645
3406 015470 104000 HLT
3407
3408                                     ;READ DATA
3409 015472 005067 163526 MRDCK3: CLR WORK3 ;CLEAR CLOCK COUNT FOR DATA WD
3410 015476 012705 026572 MOV #INBUF,R5 ;GET STARTING ADDRESS FOR DATA BUFFER
3411 015502 162705 000002 SUB #2,R5
3412 015506 012767 000025 163470 MOV #21,REPT1 ;SETUP COUNTER FOR 1ST SB BIT
3413 015514 012767 002200 163460 MOV #1152.,REPT ;SETUP COUNTER TO TRANSFER
3414                                     ;128 WORDS-9X128=1152
3415                                     ;2 CLOCKS PER 2 BITS OF DATA
3416 015522 104444 1S: RBIT ;GET 2 DATA BITS
3417 015524 104440 CLKR1 ;CLOCK MR
3418 015526 104000 HLT ;MR REG NOT CORRECT
3419 015530 104442 CLKR2 ;CLOCK MR REG
3420 015532 104000 HLT ;MR REG NOT CORRECT
3421 015534 005367 163442 DEC REPT ;DONE WITH DATA BUFFER YET?
3422 015540 001370 BNE 1S ;NO

```


H07

MAINDEC-11-DERSD-8
DERSDB.P11 TST55

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(732) 04-OCT-76 13:11 PAGE 86
MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

3423	015542	032767	000400	163416	2\$:	BIT	#BIT8,ONCEE	;DID WE ALREADY DO CRC?
3424	015550	001020				BNE	3\$;YES
3425	015552	052767	000400	163406		BIS	#BIT8,ONCEE	;NO SET CRC FLAG
3426	015560	012702	026572			MOV	#INBUF,R2	;MOVE CRC
3427	015564	062702	000440			ADD	#440,R2	;WORD TO END OF
3428	015570	012712	000000		4\$:	MOV	#0,R2	;INBUF TABLE
3429	015574	010205			5\$:	MOV	R2,R5	;GET CRC WORD
3430	015576	162705	000002			SUB	#2,R5	
3431	015602	012767	000011	163372		MOV	#9.,REPT	;SETUP TO TRANSFER 1 WD
3432	015610	000744				BR	1\$;TRANSFER CRC WD
3433	015612	104422			3\$:	MRCLK		;CLOCK MR REG
3434	015614	104420				MRCK		;CHECK MR REG
3435	015616	003601				3601		;TO EQUAL
3436	015620	104000				HLT		;3601
3437	015622	104422				MRCLK		;CLOCK MR REG
3438	015624	104420				MRCK		;CHECK MR
3439	015626	153601				153601		;TO EQUAL
3440	015630	104000				HLT		;153601
3441	015632	104422				MRCLK		;CLOCK MR REG
3442	015634	104420				MRCK		;CHECK MR
3443	015636	007601				7601		;TO EQUAL
3444	015640	104000				HLT		;7601
3445	015642	104422				MRCLK		;CLOCK MR REG
3446	015644	104420				MRCK		;CHECK MR
3447	015646	153601				153601		;TO EQUAL
3448	015650	104000				HLT		;153601

```

3449          ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3450          ;STEP INTO END OF SECTOR DEAD BAND
3451          ;EBL IS NOW ASSERTED
3452
3453 015652 012767 000010 163322 MRDCK4: MOV      #8.,REPT
3454
3455          1$: MRCLK          ;CLOCK MR REG
3456          MRCK          ;CHECK MR REG
3457          3601          ;TO EQUAL
3458          HLT          ;3601
3459          MRCLK        ;CLOCK MR REG
3460          MRCK          ;CHECK MR
3461          153601       ;REG TO
3462          HLT          ;EQUAL 153601
3463          015700 005367 163276 DEC      REPT
3464          015704 001365          BNE      1$
3465          ;NO
3466          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3467          ;SHOULD GET STROBE BUFFER
3468
3469          MRCLK          ;CLOCK MR REG
3470          MRCK          ;CHECK MR
3471          7601          ;REG TO
3472          HLT          ;EQUAL 7601
3473
3474          ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3475          ;SHOULD COMPLETE TRANSFER.
3476
3477 015716 104422          MRDCK5: MRCLK          ;CLOCK MR REG
3478 015720 022777 144270 163152 CMP      #144270, @RSCS1 ;IS RSCS1 CORRECT?
3479 015726 001401          BEQ      1$          ;YES
3480 015730 104054          HLT      !DA!DS!WC
3481 015732 005777 163146          1$: TST      @RSMC          ;DID WC GO TO 0
3482 015736 001401          BEQ      +4          ;YES
3483 015740 104010          HLT      !WC          ;WC REG SHOULD=0
3484 015742 022777 100000 163144 CMP      #100000, @RSER ;DID DCK SET?
3485 015750 001417          BEQ      3$          ;YES
3486 015752 104050          HLT      !DS!WC
3487 015754 104402 015760          TYPE     +2          ;ASCIZ <15><12>"DCK DID NOT SET "
3488 016004 004767 004206          JSR      PC, CRCTYP ;GET IC THAT FAILED AND TYPE IT
3489 016010 000241          3$: CLC
3490 016012 006167 163154          ROL      SAVEE          ;GET NEXT CRC WORD
3491 016016 103404          BCS      2$          ;DONE - BRANCH
3492 016020 004767 007542          JSR      PC, MDATA      ;SHIFT DATA PATTERN
3493 016024 000167 177060          JMP      MRDCK        ;RESTART TEST WITH NEW DATA PATTERN
3494 016030          2$:          ;DONE

```



```

3495 ;*****
3496 ;TEST 56          IGNORE FUNCTION TEST
3497 ;*****
3498 016030 104400 TST56: SCOPE
3499
3500 ;MODULE TESTED: M7759, M7770
3501 ;PUT THE DISK MAINTENANCE MODE AND SET ERROR CONDITIONS IN
3502 ;THE DRIVE ERROR REGISTER (RSER). TRY TO START A READ
3503 ;TRANSFER. THE "GO" BIT IN RSCS1 SHOULD NOT SET. MISSED
3504 ;TRANSFER ERROR (MXF) SHOULD SET IN RSCS2 WHICH IS TURN SHOULD
3505 ;CAUSE "TRE" AND "SC" TO SET IN RSCS1.
3506
3507 016032 104414 MRIFT: CLRDK ;CLEAR ALL REGISTERS
3508 016034 104430 MRIND ;SEND INDEX PULSE TO MR REG
3509 016036 104420 MRCK ;CHECK MR REG
3510 016040 022701 22701 ;TO EQUAL 22701
3511 016042 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3512 016044 012777 177777 163042 MOV #-1,RSER ;SET ERRORS
3513 016052 016777 163104 163036 MOV UNITSV,RSAS ;CLEAR ATA BIT IN RSAS
3514 ;AND ERROR BITS IN RSCS1
3515 016060 012777 027372 163020 MOV #OUTBUF,RSBA ;LOAD RSBA
3516 016066 012777 177777 163010 MOV #-1,RSWC ;LOAD RSWC
3517 016074 012777 000071 162776 MOV #71,RSXS1 ;LOAD READ FUNCTION
3518 016102 032777 000001 162770 BIT #BIT0,RSXS1 ;IS "GO" BIT ZERO?
3519 016110 001401 IS ;YES
3520 016112 104140 HLT !DS!AS ;"GO" BIT IN RSCS1 SHOULD NOT
3521 ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
3522 016114 012767 177777 163072 1$: MOV #177777,WORK ;SETUP TIMEOUT FOR MXF ERROR
3523 016122 005367 163066 5$: DEC WORK
3524 016126 000240 NOP
3525 016130 000240 NOP
3526 016132 001373 BNE 5$
3527 016134 017700 162742 MOV RSXS2,BAD ;CHECK RSCS2 FOR MXF
3528 016140 012701 001100 MOV #1100,GOOD ;GET CORRECT ANS
3529 016144 056701 163010 BIS UNNUM,GOOD ;FOR RSCS2
3530 016150 020001 CMP BAD,GOOD ;IS RSCS2 CORRECT
3531 016152 001401 BEQ 2$ ;YES
3532 016154 104000 HLT ;BAD=RSCS2 GOOD=CORRECT ANS
3533 ;MXF SHOULD BE SET IN RSCS2
3534 ;FOR A READ WAS ISSUED
3535 ;WITH ERROR BITS SET IN RSER.
3536 016156 022777 144270 162714 2$: CMP #144270,RSXS1 ;IS RSCS1 CORRECT?
3537 016164 001401 BEQ 3$ ;YES
3538 016166 104042 HLT !DS!ER ;SC AND TRE SHOULD BE SET FOR
3539 ;MXF SHOULD BE SET IN RSCS2

```



```

3564 ;*****
3565 ;TEST 57 INVALID ADDRESS ERROR (IAE) TEST
3566 ;*****
3567 016250 104400 †TST57: SCOPE
3568
3569 ;MODULE TESTED M7754, M7770
3570 ;FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
3571 ;ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN
3572 ;THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
3573 ;RSCSI WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
3574 ;DRIVE STATUS REGISTER (RSDS) AND "TRE" AND "SC" TO SET IN THE
3575 ;CONTROL REGISTER (RSCS1).
3576 016252 042767 000040 162706 BIC #BIT5,ONCEE ;CLEAR CLK CNT FLAG
3577 016260 012702 004000 MOV #4000,R2 ;LOAD R2 WITH INVALID ADDR
3578 016264 012767 016272 162516 MOV #4$,LAD ;LOOP TO HERE ON ERROR
3579 016272 104416 4$: MRDMD ;PUT DRIVE IN MAINT MODE
3580 016274 104420 MRCK ;CHECK MAINT REG
3581 016276 022701 22701 MRINT
3582 016300 104424 ;INIT MAINT MODE (CLEAR MRSP)
3583 016302 032767 000004 162656 BIT #BIT2,ONCEE ;LOOPING ON ERRORS)
3584 016310 001002 BNE 1$ ;YES
3585 016312 006102 ROL R2 ;GET INVALID ADDRESS
3586 016314 103454 BCS IADONE ;DONE FLOATING A ONE YET?
3587 016316 010277 162566 1$: MOV R2,RSDA ;LOAD RSDA WITH INVALID ADDRESS
3588 016322 012777 000071 162550 MOV #71,RSCS1 ;DO A READ TO INVALID ADDR
3589 016330 022777 002000 162556 CMP #2000,RSER ;IS RSER CORRECT?
3590 016336 001404 BEQ 2$ ;YES
3591 016340 052767 000004 162620 BIS #BIT2,ONCEE ;SET ERROR BIT
3592 016346 104044 HLT !DS!DA ;RSER SHOULD=2000 FOR
3593 ;A READ COMMAND WAS GIVEN
3594 ;TO AN ILLEGAL ADDRESS
3595 016350 042767 000004 16261C 2$: BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
3596 016356 022777 150600 162526 CMP #150600,RSDS ;DID IAE SET?
3597 016364 001404 BEQ 3$ ;YES
3598 016366 052767 000004 162572 BIS #BIT2,ONCEE ;SET ERROR BIT
3599 016374 104044 HLT !DS!DA ;RSDS SHOULD=150600 FOR
3600 ;IAE SHOULD BE SET IN RSER
3601 016376 042767 000004 162562 3$: BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
3602 016404 032777 100000 162466 BIT #BIT15,RSCS1 ;DID SC SET?
3603 016412 001004 BNE 5$ ;YES
3604 016414 052767 000004 162544 BIS #BIT2,ONCEE ;SET ERROR BIT
3605 016422 104044 HLT !DA!DS ;SC SHOULD BE SET IN RSCS1
3606 ;FOR IAE SHOULD BE SET IN RSER
3607 016424 042767 000004 162534 5$: BIC #BIT2,ONCEE ;CLEAR ERROR BIT
3608 016432 104414 CLRDK ;CLEAR ALL ERRORS
3609 016434 005777 162454 TST RSER ;DID IAE CLEAR?
3610 016440 001401 BEQ .+4 ;YES
3611 016442 104040 HLT !DS ;IAE DID NOT CLEAR
3612 016444 000712 BR 4$ ;CONTINUE
3613 016446 IADONE: ;DONE

```

```

3614 ;*****
3615 ;TEST 60 OPERATION INCOMPLETE ERROR TEST
3616 ;*****
3617 016446 104400 TST60: SCOPE
3618
3619 ;MODULE TESTED M7770
3620 ;PUT THE DISK IN MAINTENANCE MODE AND START A READ COMMAND
3621 ;THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE
3622 ;ROTATION OF THE DISK SURFACE. THE THIRD INDEX PULSE SHOULD
3623 ;CAUSE OPERATION IN COMPLETE "OPI" TO APPEAR IN THE DRIVE ERROR
3624 ;REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSDS)
3625
3626 016450 104414 MROPI: CLRDK ;CLEAR ALL DRIVE REGISTERS
3627 016452 013777 027372 162426 MOV @#OUTBUF,@RSBA ;SETUP RSBA
3628 016460 012777 177777 162416 MOV #-1,@RSC ;SETUP RSC
3629
3630 016466 104430 MRIND ;SEND INDEX PULSE TO MR REG
3631 016470 104420 MRCK ;CHECK MAINT REG
3632 016472 022701 22701 ;TO EQUAL 22701
3633 016474 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3634
3635 016476 012777 000071 162374 MOV #71,@RSCS1 ;LOAD A READ COMMAND
3636
3637 016504 104430 MRIND ;ISSUE THREE INDEX
3638 016506 104430 MRIND ;PULSES TO
3639 016510 104430 MRIND ;CAUSE OPI
3640
3641 ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
3642 016512 017700 162376 MOV @RSER,BAD ;GET RSER
3643 016516 012701 020000 MOV #20000,GOOD ;GET CORRECT ANS
3644 016522 020100 CMP GOOD,BAD ;DID OPI SET IN RSER?
3645 016524 001434 BEQ 1$ ;YES
3646 016526 104402 016532 TYPE ,.+2 ;ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
3647 016614 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
3648
3649 016616 022777 150600 162266 1$: CMP #150600,@RSDS ;DID CORRECT ERRORS SET?
3650 016624 001401 BEQ 2$ ;YES
3651 016626 104040 HLT !DS ;RSDS SHOULD=150600 BECAUSE
3652 ;OF OPI ERROR IN RSER
3653 016630 022777 144270 162242 2$: CMP #144270,@RSCS1 ;DID SC AND TRE SET IN RSCS1?
3654 016636 001401 BEQ MROPIA ;YES
3655 016640 104050 HLT !DS!WC ;SC AND TRE SHOULD SET IN RSCS1
3656 ;BECAUSE OF ERROR IN RSER
3657 016642 104414 MROPIA: CLRDK ;CLEAR ALL ERRORS
3658 016644 005777 162244 TST @RSER ;DID OPI CLEAR IN RSER
3659 016650 001437 BEQ 1$ ;YES
3660 016652 104402 016656 TYPE ,.+2 ;ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
3661 016746 104040 HLT !DS ;RSER SHOULD=0
3662 016750 022777 010600 162134 1$: CMP #10600,@RSDS ;DID ERROR BITS CLEAR IN RSDS
3663 ;BY SETTING CLR BIT IN RSCS2
3664 016756 001401 BEQ ,+4 ;YES
3665 016760 104040 HLT !DS ;RSDS SHOULD=10600

```



```

3666 :*****
3667 :TEST 61 PARITY ERROR TEST
3668 :*****
3669 016762 104400 TST61: SCOPE
3670
3671 :MODULES TESTED: M7754, M7770
3672 :SET "PAT" BIT IN RSCS2. WRITE A DRIVE REGISTER. "PAR" SHOULD SET IN
3673 :THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS
3674 :AND 'SC' TO SET IN RSCS1.
3675
3676 016764 104414 MRPAR: CLRDK ;CLEAR ALL REGISTERS
3677 016766 042767 000040 162172 BIC #BITS,ONCEE ;CLEAR CLK CNT FLAG
3678 016774 104430 MRIND ;SEND INDEX PULSE TO MR REG
3679 016776 104420 MRCK ;CHECK MAINT TO
3680 017000 022701 22701 ;EQUAL 22701
3681 017002 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3682 017004 052777 000020 162070 BIS #BIT4,RSRCS2 ;SET THE "PAT" BIT.
3683 017012 012777 000077 162070 MOV #77,RSRSDA ;BY WRITING INTO THIS REGISTER,
3684 ;PAR SHOULD SET IN RSER
3685 017020 022777 000010 162066 CMP #10,RSRER ;DID PAR SET?
3686 017026 001401 BEQ +4 ;YES
3687 017030 104040 HLT !DS ;"PAR" IN RSER SHOULD BE SET FOR
3688 ;THE "PAT" BIT WAS SET IN RSCS2
3689 ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
3690 017032 022777 104200 162040 CMP #104200,RSRCS1 ;DID PAR CAUSE SC TO SET?
3691 017040 001401 BEQ +4 ;YES
3692 017042 104044 HLT !DS!DA ;SC SHOULD BE SET IN RSCS1 FOR
3693 ;PAR SHOULD BE SET IN RSER
3694 017044 022777 000077 162036 CMP #77,RSRSDA ;DID RSDA GET LOADED?
3695 017052 001401 BEQ +4 ;YES
3696 017054 104004 HLT !DA ;RSDA SHOULD=77 FOR PAT
3697 ;BIT WAS SET WHEN PROGRAM
3698 ;TRIED TO WRITE INTO RSDA
3699 017056 104414 CLRDK ;CLEAR ALL ERRORS
3700 017060 022777 004200 162012 CMP #4200,RSRCS1 ;DID ERRORS CLEAR?
3701 017066 001401 BEQ +4 ;YES
3702 017070 104044 HLT !DS!DA ;SC DID NOT CLEAR BY USING
3703 ;THE "CLR" BIT IN RSCS2
3704 017072 005777 162016 TST RSRER ;DID PAR CLEAR?
3705 017076 001401 BEQ +4 ;YES
3706 017100 104044 HLT !DS!DA ;PAR DID NOT CLEAR BY USING
3707 ;THE CLR BIT IN RSCS2

```

```

3708 :*****
3709 :TEST 62 MAINTENANCE MODE INTERRUPT TEST
3710 :*****
3711 017102 104400 TST62: SCOPE
3712
3713 :MODULE TESTED M7771
3714 :IN THIS TEST THE INTERRUPT ENABLE BIT IS SET (I.E.).
3715 :A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMR"
3716 :ERROR IS CREATED WHILE THE FIRST SECTOR IS BEING WRITTEN
3717 :THIS SHOULD CAUSE THE DRIVE TO INTERRUPT AFTER THE FIRST
3718 :SECTOR IS WRITTEN AND THE TRANSFER TO TERMINATE.
3719
3720 017104 012767 000002 162032 MREX: MOV #2,FLAG2
3721 017112 104414 CLRDK ;CLEAR DRIVE REGISTERS
3722 017114 012737 000200 177776 MOV #200,SPS ;SETUP FOR INTERRUPT
3723 017122 012706 000500 MOV #500,SP
3724 017126 052767 000040 162032 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
3725 017134 042767 000600 162024 BIC #600,ONCEE ;CLEAR FLAG BITS
3726 017142 104430 MRIND ;SEND INDEX PULSE TO MR REG
3727 017144 104420 MRCK ;CHECK MR REG
3728 017146 022701 22701 ;TO EQUAL 22701
3729 017150 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3730 ;BY SENDING 2 CLOCK PULSES
3731
3732 :FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
3733 :DATA BUFFER WORDS ARE :A WORD OF ALL 0'S - ALL 1'S
3734 : FLOATING 1'S PATTERN (16 WORDS)
3735 : A PATTERN OF 146314 (110 WORDS)
3736 017152 012702 026572 MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
3737 017156 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
3738 017160 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
3739 017164 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
3740 017166 000261 SEC ;A PATTERN OF FLOATING ONES
3741 017170 006103 15: ROL R3 ;GET PATTERN
3742 017172 103402 BCS 25 ;DONE GET OUT
3743 017174 010322 MOV R3,(R2)+ ;FILL BUFFER
3744 017176 000774 BR 15 ;CONT
3745 017200 012703 000156 25: MOV #110,R3 ;FILL REMAINING PORTION OF
3746 017204 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
3747 017210 010422 35: MOV R4,(R2)+ ;LOAD BUFFER
3748 017212 005303 DEC R3 ;DONE YET?
3749 017214 001375 BNE 35 ;NO
3750
3751 ;SETUP CONTROLLER TO TRANSFER 256 WORDS OF DATA (2 SECTORS)
3752 017216 012777 020024 161704 MOV #INTMR,RSVEC ;SETUP INTERRUPT VECTOR
3753 017224 012777 000340 161700 MOV #340,RSVCPS
3754 017232 012777 026572 161646 MOV #INBUF,RSBA ;LOAD BUS ADDR REG
3755 017240 012777 177400 161636 MOV #177400,RSWC ;LOAD WORD COUNT REG
3756 017246 012777 000161 161624 MOV #161,RSCSI ;LOAD WRITE COMMAND I/E
3757 017254 104446 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3758 ;TO CLEAR OUT COUNTERS AND REGISTERS
3759 ;THAT OTHERWISE COULD NOT BE CLEARED.
3760 017256 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3761 017260 104450 SPASS ;CLOCK MR REG SP = 1

```



```

3762                ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
3763 017262 104430  MRIND
3764 017264 104420  MRCK                ;CHECK MR REG TO EQUAL
3765 017266 020501  20501                ;20501 FOR A
3766 017270 104000  HLT                ;WRITE COMD
3767
3768                ;STEP THRU RESYNC PERIOD
3769
3770 017272 012767 001000 161702  MOV      #512.,REPT
3771 017300 052767 000040 161660  MREX1:  BIS      #BITS,ONCEE                ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
3772 017306 104422                MRCLK                ;CLOCK MR REG
3773 017310 104420                MRCK                ;CHECK FOR
3774 017312 070501                70501                ;CORRECT DATA
3775 017314 104000                HLT                ;MR = BAD GOOD = CORRECT DATA
3776 017316 104422                MRCLK                ;CLOCK MR REG
3777 017320 104420                MRCK                ;CHECK FOR
3778 017322 020501                20501                ;CORRECT DATA
3779 017324 104000                HLT                ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
3780 017326 005367 161650  DEC      REPT                ;FINISH LOOPING
3781 017332 001365                BNE      MREX1                ;THROUGH RESYNC PERIOD
3782
3783                ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
3784                ;SP=0 EQUALS SECTOR PULSE
3785 017334 104422                MRCLK                ;CLOCK MR REG
3786 017336 104420                MRCK                ;MR SHOULD
3787 017340 070101                70101                ;EQUAL 70101
3788 017342 104000                HLT                ;MR=BAD GOOD=CORRECT ANS
3789 017344 104422                MRCLK                ;CLOCK MR REG
3790 017346 104420                MRCK                ;CHECK MR
3791 017350 020101                20101                ;TO EQUAL 20101
3792 017352 104000                HLT                ;MR=BAD GOOD=CORRECT ANS
3793
3794                ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE
3795
3796 017354 012767 000077 161620  MREX2:  MOV      #63.,REPT                ;CLOCK MR REG
3797 017362 104422                MRCLK                ;CHECK MR REG
3798 017364 104420                MRCK                ;TO EQUAL 71501
3799 017366 071501                71501                ;MR=BAD GOOD=CORRECT ANS
3800 017370 104000                HLT                ;CLOCK MR REG
3801 017372 104422                MRCLK                ;CHECK MR REG
3802 017374 104420                MRCK                ;TO EQUAL 21501
3803 017376 021501                21501                ;MR=BAD GOOD=CORRECT ANS
3804 017400 104000                HLT                ;DONE YET
3805 017402 005367 161574  DEC      REPT                ;NO LOOP
3806 017406 001365                BNE      MREX2

```


E08

M3INDEC-11-DERSD-B
DERSDB.P11

TST62

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
MAINTENANCE MODE INTERRUPT TEST

MACY11 27(732) 04-OCT-76 13:11 PAGE 96

3853	017532	012767	002167	161454		MOV	#1143.,WORK		
3854									:DOING A 1 SECTOR TRANSFER 127 WORDS
3855									:18 BITS PER WORD-CLOCK LOOPS
3856									:TAKE CARE OF 2 BITS AT A TIME
3857									:127 TIMES 9 EQUALS 1143 LOOPS
3858	017540	042767	000200	161420		BIC	#BIT7,ONCEE		:TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
3859	017546	052767	000100	161412		BIS	#BIT6,ONCEE		:CLEAR LAST WORD FLAG
3860	017554	104432			1\$:	XBIT			:SET 1ST TRANSFER WORD FLAG
3861	017556	104434				CLKD1			:GET 2 BITS OF DATA
3862									:SEND FIRST CLOCK PULSE
3863									:AND CALCULATE MR REG
3864	017560	104000				HLT			:FOR CORRECT DATA (MWD+MWD)
3865	017562	104436				CLKD2			:MR REG NOT CORRECT
3866									:SEND 2ND CLOCK PULSE TO
3867									:COMPLETE TRANSFER OF 2 BITS
3868									:CALCULATE CORRECT ANS FOR
3869	017564	104000				HLT			:MR REG (MWD+MWD)
3870	017566	032767	000200	161372		BIT	#BIT7,ONCEE		:MR=BAD GOOD=CORRECT ANS
3871	017574	001015				BNE	2\$:ON LAST WORD YET
3872	017576	032767	000400	161362		BIT	#BIT8,ONCEE		:YES
3873	017604	001043				BNE	3\$:ON CRC WORD YET?
3874	017606	005367	161402			DEC	WORK		:YES
3875	017612	001360				BNE	1\$:DONE WITH 127 WORDS?
3876									:NO
3877	017614	052767	000200	161344		BIS	#BIT7,ONCEE		:SET LAST WORD FLAG
3878	017622	012767	000012	161364		MOV	#10.,WORK		:SET UP TO TRANSFER LAST WORD
3879	017630	005367	161360		2\$:	DEC	WORK		:DONE YET
3880	017634	001347				BNE	1\$		
3881									
3882	017636	052767	000400	161322		BIS	#BIT8,ONCEE		:SET TRANSFERRING CRC WORD
3883	017644	042767	000200	161314		BIC	#BIT7,ONCEE		:CLEAR LAST WORD FLAG
3884									
3885									:GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
3886									:EXC SHOULD THEN BE ASSERTED
3887									
3888	017652	012777	177777	161234		MOV	#-1,RSER		
3889	017660	004767	003570			JSR	PC,GENCRC		:GENERATE CRC WORD
3890									:AND LEAVE IN "WORK"
3891	017664	012702	026572			MOV	#INBUF,R2		:GO TO END
3892	017670	062702	000400			ADD	#400,R2		:OF DATA BUFFER
3893	017674	016712	161314			MOV	WORK,R2		:LOAD CRC WORD
3894	017700	010205				MOV	R2,R5		:RESET POINTER FOR
3895	017702	162705	000002			SUB	#2,R5		:R5 FOR CRC WD
3896	017706	012767	000012	161300		MOV	#10.,WORK		:SETUP TO XFER CRC
3897	017714	005367	161274		3\$:	DEC	WORK		:DONE YET?
3898	017720	001315				BNE	1\$:NO

```

3899          ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN
3900 017722 104422          MRCLK          ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
3901 017724 104420          MRCK           ;CHECK MR REG
3902 017726 153501          153501        ;TO EQUAL 103501
3903 017730 104000          HLT           ;MR REG=BAD GOOD=CORRECT ANS
3904
3905          ;LOOP 6 TIMES
3906 017732 012767 000006 161242 ;LOOP 6 TIMES          MOV          #6,REPT
3907 017740 104422          4$: MRCLK          ;CLOCK MR REG
3908 017742 104420          MRCK           ;CHECK MR REG
3909 017744 003501          3501        ;TO EQUAL 53501
3910 017746 104000          HLT           ;MR=BAD GOOD=CORRECT ANS
3911 017750 104422          MRCLK          ;CLOCK MR REG
3912 017752 104420          MRCK           ;CHECK MR REG
3913 017754 153501          153501        ;TO EQUAL 103501
3914 017756 104000          HLT           ;MR=BAD GOOD=CORRECT ANS
3915 017760 005367 161216          DEC          REPT
3916 017764 001365          BNE          4$          ;DONE LOOPING YET?
3917
3918          ;FINISH UP
3919 017766 104422          MRCLK          ;CLOCK MR REG
3920 017770 104420          MRCK           ;CHECK MR REG
3921 017772 003501          3501        ;TO EQUAL 3501
3922 017774 104000          HLT           ;MR REG=BAD GOOD=CORRECT ANS
3923 017776 104422          MRCLK          ;CLOCK MR REG
3924 020000 104420          MRCK           ;CHECK MR REG
3925 020002 151501          151501        ;TO EQUAL 151501
3926 020004 104000          HLT           ;MR=BAD GOOD=CORRECT ANS
3927
3928          ;TRANSFER SHOULD NOW BE COMPLETE
3929 020006 104422          MRCLK          ;CLOCK MR REG
3930 020010 104420          MRCK           ;CHECK MR REG
3931 020012 002701          2701        ;TO EQUAL 2701
3932 020014 104000          HLT           ;MR=BAD GOOD=CORRECT ANS
3933 020016 000240          NOP          ;STALL FOR TIME
3934 020020 104050          HLT           ;SHOULD NEVER GET HERE
3935 020022 000424          BR          !WC!DS      ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED.
3936
3937
3938          ;NOW TEST CONTROLLER
3939
3940 020024 022777 144260 161046 INTMR: CMP          #144260,RS1  ;IS CS1 CORRECT?
3941 020032 001401          BEQ          .+4        ;YES
3942 020034 104014          HLT          !DA!WC      ;YES
3943 020036 022777 177610 161040 5$: CMP          #177610,RSWC  ;IS WC REG CORRECT?
3944 020044 001401          BEQ          .+4        ;YES
3945 020046 104010          HLT          !WC        ;WC SHOULD BE = TO 177610
3946 020050 022777 000004 161036 CMP          #4,RSER      ;DID RMR SET IN RSER
3947 020056 001401          BEQ          .+4        ;YES
3948 020060 104050          HLT          !DS!WC      ;RSER SHOULD = 4
3949 020062 022777 000001 161020 CMP          #1,RS1DA     ;DOES RSDA=1
3950 020070 001401          BEQ          .+4        ;YES
3951 020072 104004          HLT          !DA        ;RSDA SHOULD=1
3952 020074 000240          INTMR1: NOP          ;DONE

```



```

3953 ;*****
3954 ;TEST 63 DISK ADDRESS OVERFLOW TEST
3955 ;*****
3956 020076 104400 TST63: SCOPE
3957
3958 ;MODULES TESTED: M7754, M7771, M7770
3959 ;SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77
3960 ;TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER
3961 ;(LBT) BIT TO SET IN THE RSDS REGISTER.
3962
3963 020100 104414 MRAOE: CLRDK ;CLEAR ALL REGISTERS
3964 020102 012706 000500 MOV #500,SP ;SETUP STACK POINTER
3965 020106 104430 MRIND ;SEND INDEX PULSE TO MR REG
3966 020110 104420 MRCK ;CHECK MAINT REG
3967 020112 022701 22701 ;TO EQUAL 22701
3968 020114 104424 MRINT ;INITIALIZE MAINT REG BY SENDING
3969 ;2 CLOCK PULSES (CLEAR MRSP)
3970 020116 012777 007777 160764 MOV #7777,RSDA ;SETUP DISK ADDRESS
3971 020124 012777 177400 160752 MOV #-400,RSWC ;SETUP FOR A 2 SECTOR TRANSFER
3972 020132 012777 027372 160746 MOV #OUTBUF,RSBA ;GET OUTPUT BUFFER
3973
3974 ;SETUP BUFFER WITH ALL ONES
3975 020140 012705 027372 MOV #OUTBUF,R5 ;GET STARTING ADDRESS OF OUTBUF
3976 020144 012767 000400 161030 MOV #400,REPT ;LOAD 2 SECTORS
3977 020152 012725 177777 1S: MOV #-1,(R5)+ ;WITH WORDS
3978 020156 005367 161020 DEC REPT ;OF ALL ONES
3979 020162 001373 BNE 1S
3980
3981 020164 012777 000061 160706 MOV #61,RSCSI ;LOAD WRITE COMMAND
3982 020172 104430 MRIND ;SET INDEX PULSE
3983
3984 ;SUPPLY CLOCKS TO STEP THROUGH A TRACK
3985
3986 020174 012767 000003 161000 MOV #3,REPT
3987 020202 012704 160000 5S: MOV #57344.,R4 ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
3988 020206 012702 000011 MOV #11,R2 ;(3 X 57344 = 172032)
3989 020212 012703 000001 MOV #1,R3
3990 020216 010277 160702 2S: MOV R2,RSMR
3991 020222 010377 160676 MOV R3,RSMR
3992 020226 005304 DEC R4
3993 020230 001372 BNE 2S
3994 020232 005367 160744 DEC REPT
3995 020236 001361 BNE 5S
3996
3997 ;CAUSE "LBT IN RSDS TO SET
3998
3999 020240 104422 MRCLK ;CLOCK AN 11 AND A 1 INTO RSMR
4000 020242 104426 DSCK ;CHECK MR
4001 020244 012400 12400 ;TO EQUAL 12400
4002 020246 104000 HLT ;LBT SHOULD BE SET IN RSDS

```

```

4003
4004
4005      ;ASSERT MAINTENANCE INDEX PULSE TO RESET DRIVE
4006      ;FOR THE SECOND REVOLUTION
4007 020250 104430      MRIND      ;ASSERT MAINT INDEX PULSE
4008 020252 005067 160716 CLR      MCCNT      ;CLEAR THE CLOCK COUNTER
4009 020256 104420      MRCK      ;CHECK MR REG
4010 020260 002501      2501      ;TO EQUAL 2501. SHOULD STILL BE WRITING
4011 020262 104000      HLT
4012
4013      ;SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS04 RESYNC PERIOD
4014 020264 012767 001000 160710 MOV      #512.,REPT ;CLOCK COUNT TO STEP THRU RESYNC
4015 020272 104422      4$: MRCLK      ;2ND REVOLUTION
4016 020274 104420      MRCK      ;CHECK MR
4017 020276 052501      52501     ;TO EQUAL 52501
4018 020300 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
4019 020302 104422      MRCLK      ;CLOCK MR REG
4020 020304 104420      MRCK      ;CHECK MR
4021 020306 002501      2501      ;REG TO
4022 020310 104000      HLT      ;EQUAL 2501
4023 020312 005367 160664 DEC      REPT
4024 020316 001365      BNE      4$      ;LOOP TILL DONE
4025
4026      ;SUPPLY 2 CLOCKS TO CAUSE THE SECTOR PULSE TO APPEAR IN
4027      ;THE MR REGISTER AND THE "AOE" ERROR TO APPEAR IN
4028      ;THE RSER REGISTER
4029
4030 020320 104422      AOECK: MRCLK
4031 020322 104422      MRCLK
4032 020324 104420      MRCK
4033 020326 022301      22301
4034 020330 104000      HLT
4035 020332 022777 001000 160554 CMP      #1000,RSER
4036 020340 001401      BEQ      1$
4037 020342 104040      HLT      !DS
4038 020344 022777 152600 160540 1$: CMP      #152600,RSDS
4039 020352 001401      BEQ      2$
4040 020354 104040      HLT      !DS
4041
4042 020356 104414      2$: CLDK
4043 020360 005777 160530 TST      RSER
4044 020364 001401      BEQ      3$
4045 020366 104040      HLT      !DS
4046 020370 022777 010600 160514 3$: CMP      #10600,RSDS
4047 020376 001401      BEQ      +4
4048 020400 104040      HLT      !DS
4049
;ERR & ATA SHOULD BE SET IN RSDS
;BECAUSE OF AOE ERROR IN RSER
;CLEAR ERROR
;DID ERROR CLEAR?
;YES
;AOE DID NOT CLEAR BY SETTING CLR IN RSCS2
;DID ERRORS CLEAR
;YES
;ERR AND ATA & LBT SHOULD ALL BE CLEARED
;FOR CLR WAS SET IN RSCS2

```



```

4050 ;MAINTENANCE MODE VERIFY TEST
4051 ;-----DANGER---THIS TEST DESTROYS DATA ON DISKS--DANGER
4052 ;THIS TEST WILL ONLY RUN IF SWITCH 11 IS SET IN THE "SWITCH
4053 ;REGISTER" FOR IT WILL ACTUALLY WRITE DATA INTO THE DISK. IT
4054 ;WILL WRITE ONE TRACK OF ALL ONES. THE PROGRAM THEN GOES BACK
4055 ;TO THE MAINT WRITE TEST AND WRITES ONE SECTOR OF DATA (ZER)'S, ONES, FLOATING
4056 ;ONES AND FILLS THE REMAINDER OF SECTOR WITH A PATTERN OF 146314)
4057 ;THE DRIVE IS THEN TAKEN OUT OF
4058 ;"MAINTENANCE MODE" AND THE TRACK IS THEN READ. THE TRACK
4059 ;SHOULD CONTAIN ALL ONES.

```

```

4060 ;*****
4061 ;TEST 64 MAINTENANCE MODE VERIFY TEST
4062 ;*****
4063 ;TST64: SCOPE

```

```

4064 020402 104400
4065 ;MODULE TESTED G182

```

4068	020404	032767	004000	157156	MRVR:	BIT	#BIT11,SWR	:DO THIS TEST?
4069	020412	001002				BNE	3\$:YES
4070	020414	000137	021130			JMP	@#INFTST	:NO
4071	020420	005067	160520		3\$:	CLR	FLAG2	:SET VERIFY TEST FLAG
4072	020424	104414				CLRDK		:CLEAR ALL DRIVES
4073	020426	012767	177777	160574		MOV	#177777,WORK5	:STALL TO
4074	020434	005367	160570		4\$:	DEC	WORK5	:RESYNC DRIVE
4075	020440	001375				BNE	4\$:TIMING LOGIC
4076	020442	042767	000040	160516		BIC	#BITS,ONCEE	:CLEAR CLK CNT
4077	020450	012777	160000	160426		MOV	#-20000,@R5WC	:WRITE ONE TRACK - 8K WDS
4078	020456	012767	177777	006106		MOV	#177777,INBUF	:WRITE A PATTERN 12525
4079	020464	052777	000010	160410		BIS	#BIT3,@R5CS2	:SET BAI BIT
4080	020472	012777	026572	160406		MOV	#INBUF,@R5BA	:SET DATA WD
4081	020500	012767	177777	160474		MOV	#177777,REPT	:SETUP WAIT LOOP
4082	020506	012777	000061	160364		MOV	#61,@R5CS1	:GO WRITE
4083	020514	105777	160360		1\$:	TSTB	@R5CS1	:DONE YET?
4084	020520	100404				BMI	2\$:YES
4085	020522	005367	160454			DEC	REPT	:DECREMENT COUNTER WAITING
4086	020526	001372				BNE	1\$:FOR READY
4087	020530	104000				HLT		:READY NEVER CAME UP
4088	020532	005777	160342		2\$:	TST	@R5CS1	:ANY ERRORS?
4089	020536	100002				BPL	MRVR1	:NO
4090	020540	104050				HLT	!DS!WC	:STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
4091	020542	000433				BR	TBDIA	:TYPE MESSAGE

```

4092 020544 104414 MRVR1: CLRDK ;CLEAR ALL REGISTERS
4093 020546 012777 160000 160330 MOV #-20000, @RSWC ;SETUP WC
4094 020554 052777 000010 160320 BIS #BIT3, @RSCS2 ;SET BAI
4095 020562 012777 026572 160316 MOV #INBUF, @RSBA ;SETUP RSBA
4096 020570 012767 177777 160404 MOV #177777, REPT ;SETUP WAIT LOOP
4097 020576 012777 000051 160274 MOV #51, @RSCS1 ;DO A WRITE CHECK TO VERIFY DISK
4098 020604 105777 160270 1S: TSTB @RSCS1 ;TEST
4099 020610 100404 BMI 2S ;FOR READY TO COME BACK
4100 020612 005367 160364 DEC REPT ;WAIT
4101 020616 001372 BNE 1S ;
4102 020620 104000 HLT ;READY NEVER CAME BACK
4103 020622 005777 160252 2S: TST @RSCS1 ;ANY ERRORS?
4104 020626 100032 BPL MRVRR ;NO
4105 020630 104050 HLT !DS!WC ;STOP HERE WC FAILED
4106 ;GO TO DZRSB DIAG
4107 ;BEFORE TRYING TO DEBUG THIS TEST
4108 020632 TBDIA:
4109 020632 104402 020636 TYPE ;ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
4110 020714 000137 011456 MRVRR: JMP @#MRWRT ;GO WRITE IN MAINTENANCE MODE
4111 ;NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
4112
4113 020720 104414 MRVR2: CLRDK ;CLEAR ALL REGISTERS
4114 020722 012767 177777 160264 MOV #177777, WORK ;STALL - TO RESPONSE
4115 020730 005367 160260 3S: DEC WORK ;INDEX PULSE
4116 020734 001375 BNE 3S ;ON DRIVE
4117 020736 012777 160000 160140 MOV #-20000, @RSWC ;SETUP WC FOR 1 TRACK
4118 020744 052777 000010 160130 BIS #BAI, @RSCS2 ;SET BAI
4119 020752 012777 026572 160126 MOV #INBUF, @RSBA ;SETUP RSBA
4120 020760 012767 177777 005604 MOV #177777, INBUF ;SETUP FOR COMPARE
4121 020766 012777 000051 160104 MOV #51, @RSCS1 ;DO A WRITE CHECK
4122 020774 105777 160100 1S: TSTB @RSCS1 ;TEST FOR
4123 021000 100375 BPL 1S ;READY TO COME BACK
4124 021002 032777 040000 160072 BIT #WCE, @RSCS2 ;DID WCE SET?
4125 021010 001442 BEQ 2S ;NO
4126 021012 104402 021016 TYPE ;ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
4127 021112 104040 HLT !DS ;
4128 021114 000404 BR 4S ;GET OUT
4129 021116 005777 157756 2S: TST @RSCS1 ;ANY ERRORS?
4130 021122 100001 BPL 4S ;NO
4131 021124 104040 HLT !DS ;SHOULD NOT BE ANY ERRORS
4132 ;TRY THE DZRSB DIAGNOSTIC
4133 021126 000240 4S: NOP
4134
4135 021130 000137 002116 INFTST: JMP @#TRYNX ;GET NEXT DRIVE

```


K08

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
\$DONE - BELL AND SCOPE ROUTINE

MACY11 27(732) 04-OCT-76 13:11 PAGE 102

```

4136                                     .SBTTL          $DONE - BELL AND SCOPE ROUTINE
4137
4138 021134 104400          DONE:  SCOPE          ; TERMINATING SCOPE FOR LOOPING
4139 021136 062767 000001 157642      ADD      #1,PCNT+2      ; ADD 1 TO THE PASS COUNT
4140 021144 005567 157634          ADC      PCNT          ; MAKE IT DOUBLE PREC.
4141 021150 032737 002000 177570      BIT      #SW10,#SWR    ; RING THE BELL?
4142 021156 001004          BNE      4$           ; NO!
4143 021160 104402 021164          TYPE     .+2          ; .ASCIZ <BELL><177>
4144 021170 013700 000042          4$:  MOV     @#42,R0    ; GET MONITOR ADDRESS
4145 021174 001404          BEQ     3$           ; IF NONE
4146 021176 004710          JSR     7,(0)        ; GO TO MONITOR
4147 021200 000240 000240 000240      240,240,240          ; SAVE ROOM FOR ACT11
4148 021206 000167 000002          3$:  JMP     MULSYS    ; RETURN
4149
4150 021212 000000          .TBIT:  0           ; T BIT FLAG
4151
4152                                     ;MULTI DRIVE SYSTEM?
4153
4154                                     MULSYS:
4155 021214 104402 021220          TYPE     .+2          ; .ASCIZ <15><12>"END OF PASS"
4156 021236 005067 157546          CLR     LAD
4157 021242 005067 157532          CLR     ICNT
4158 021246 032767 000020 157712      BIT     #BIT4,ONCEE  ; MULTI DRVIE?
4159 021254 001002          BNE     1$           ; NO
4160 021256 000137 001672          JMP     @#MULTII     ; YES
4161 021262 000137 002322          1$:  JMP     @#NOWGO  ; TEST ONLY ONE DRIVE
4162
4163                                     ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
4164
4165 021266 032767 000004 157672      NOPERR: BIT     #BIT2,ONCEE ; WERE WE HERE BEFORE?
4166 021274 001031          BNE     1$           ; YES
4167 021276 052767 000004 157662      BIS     #BIT2,ONCEE  ; SET BEEN HERE BEFORE FLAG
4168 021304 104402 021310          TYPE     .+2          ; .ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
4169 021352 016746 157636          MOV     WORK,-(6)    ; PUT WORK ON STACK
4170 021356 104406          TYPES
4171 021360 000207          1$:  RTS     PC      ; TYPE STACK IN OCTAL - SUPRESS

```

4172	021362	104402	021376		CHG:	TYPE	REGCHG		:TYPE MESSAGE
4173	021366	016746	157622			MOV	WORK,-(6)		:PUT WORK ON STACK
4174	021372	104406				TYPES			:TYPE STACK IN OCTAL - SUPRESS
4175	021374	000207				RTS	PC		
4176									
4177	021376	044103	047101	042507	REGCHG:	.ASCIZ	"CHANGED WITH NO-OP FUNCTION "		
4178	021404	020104	044527	044124					
4179	021412	047040	026517	050117					
4180	021420	043040	047125	052103					
4181	021426	047511	020116	000					
4182									
4183	021433	015	051012	051115	TRMR:	.ASCIZ	<15><12>"RMR DID NOT SET BY WRITING INTO "		
4184	021440	020040	044504	020104					
4185	021446	047516	020124	042523					
4186	021454	020124	054502	053440					
4187	021462	044522	044524	043516					
4188	021470	044440	052116	020117					
4189	021476	000							
4190		021500				.EVEN			
4191									
4192	021500	104422			.MRINT:	MRCLK			:CLOCK THE MAINT REG WITH AN 11 AND A 1
4193	021502	104422				MRCLK			:SAME
4194	021504	000002				RTI			:RETURN
4195									
4196	021506	012777	000011	157410	.MRCLK:	MOV	#11,RSMR		:CLOCK THE
4197	021514	012777	000001	157402		MOV	#1,RSMR		:MAINT REG
4198	021522	062767	000001	157446		ADD	#1,MCCNT+2		:ADD 1 TO CLOCK COUNT
4199	021530	005567	157440			ADC	MCCNT		:MAKE DOUBLE PRECISION
4200	021534	000002				RTI			
4201									
4202	021536	017700	157362		.MRCK:	MOV	RSMR,BAD		:GET THE CONTENTS OF RSMR
4203	021542	017601	000000			MOV	2(SP),GOOD		:GET THE CORRECT ANSWER
4204	021546	062716	000002			ADD	#2,(SP)		:UPDATE THE RETURN ADDRESS FOR AN ERROR
4205	021552	020100				CMP	GOOD,BAD		:IS THE MR REG CORRECT?
4206	021554	001002				BNE	1\$:NO EXIT
4207	021556	062716	000002		1\$:	ADD	#2,(SP)		:UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4208	021562	000002				RTI			:RETURN
4209									
4210									
4211	021564	012777	000021	157332					:SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4212	021572	012777	000001	157324	.MRIND:	MOV	#21,RSMR		:SEND INDEX
4213	021600	000002				MOV	#1,RSMR		:PULSE TO MR REG
4214	021602	017700	157304			RTI			
4215	021606	017601	000000		.DSCK:	MOV	RSDS,BAD		:GET THE CONTENTS OF RSDS
4216	021612	062716	000002			MOV	2(SP),GOOD		:GET THE CORRECT ANS
4217	021616	020100				ADD	#2,(SP)		:UPDATE THE RETURN ADDR FOR AN ERROR
4218	021620	001002				CMP	GOOD,BAD		:IS RSDS CORRECT
4219	021622	062716	000002			BNE	1\$:NO EXIT
4220	021626	000002			1\$:	ADD	#2,(SP)		:UPDATE RETURN ADDR TO SKIP THE HLT FOR CORRECT ANS
						RTI			


```

4221
4222
4223
4224
4225
4226
4227
4228 021630 032767 000100 157330 .XBIT: BIT #BIT6,ONCEE ;1ST 2 BITS OF 1ST WORD?
4229 021636 001427 BEQ 2$ ;NO
4230 021640 012767 000001 157300 MOV #1,LSTEV ;SET LAST EVEN BIT TRANSFERRED TO A 1
4231 021646 012767 000001 157274 MOV #1,LSTOD ;SET LAST ODD BIT TRANSFERRED TO A 1
4232 ;THIS SETS UP THE SYNC 1 BITS AT END OF PREAMBLE
4233 ;FOR THE TOP AND BOTTOM
4234 ;BITS IN THE MR REGISTER
4235 021654 042767 000100 157304 BIC #BIT6,ONCEE ;CLEAR 1ST WORD TRANSFER FLAG
4236 021662 005067 157320 4$: CLR CLKCNT ;CLEAR CLOCK COUNTER AT START OF EACH WORD
4237 021666 032767 000400 157272 BIT #BIT8,ONCEE ;CRC WORD BEING WRITTEN?
4238 021674 001042 BNE 1$ ;YES
4239 021676 005067 157252 CLR NOWOD ;NO, LOAD EVEN
4240 021702 005067 157244 CLR NOWEV ;AND ODD WITH 0 FOR BITS 16 & 17 IN RSD4 DATA WORD.
4241 021706 012767 000010 157310 6$: MOV #8.,WORK3 ;8 LOOPS FOR REMAINING 16 BITS OF WORD
4242 021714 000002 RTI
4243 021716 016767 157232 157224 2$: MOV NOWOD,LSTOD
4244 021724 016767 157222 157214 MOV NOWEV,LSTEV ;SAVE LAST 2 BITS TRANSFERRED
4245 021732 005767 157266 TST WORK3 ;DONE WITH WORD YET?
4246 021736 001004 BNE 3$ ;NO
4247 021740 062705 000002 ADD #2,R5 ;UPDATE BUFFER WD
4248 021744 011504 MOV (R5),R4 ;GET DATA WD
4249 021746 000745 BR 4$ ;GET BITS 16 & 17
4250 021750 005067 157200 3$: CLR NOWOD ;CLEAR PRESENT ODD BIT
4251 021754 006104 ROL R4 ;GET NEXT ODD DATA BIT
4252 021756 006167 157172 ROL NOWOD ;SAVE IT IN ODD BIT
4253 021762 005067 157164 CLR NOWEV ;CLEAR PRESENT EVEN BIT
4254 021766 006104 ROL R4 ;GET NEXT EVEN BIT
4255 021770 006167 157156 ROL NOWEV ;SAVE IT IN EVEN BIT
4256 021774 005367 157224 DEC WORK3 ;KEEP COUNT OF BITS IN THE WORD
4257 022000 000002 RTI ;RETURN
4258
4259 ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS. 0 & 1 ARE ALWAYS 0
4260 022002 005067 157146 1$: CLR NOWOD ;GET BITS 17
4261 022006 006104 ROL R4 ;AND 16
4262 022010 006167 157140 ROL NOWOD ;FOR CRC WORD
4263 022014 005067 157132 CLR NOWEV
4264 022020 006104 ROL R4
4265 022022 006167 157124 ROL NOWEV
4266 022026 000727 BR 6$ ;CONTINUE

```

```

4267 ;CLOCK ROUTINE (1ST OF TWO) WHICH IS USED TO CLOCK TWO BITS OF
4268 ;DATA TO THE DRIVE AT A TIME. THIS ROUTINE ALSO CHECKS THE PREVIOUS
4269 ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
4270 ;THE MWDT BIT (BIT 14 IN THE MR REG) AND MWDB BIT (BIT 12 IN THE MR REG) SHOULD BE IN
4271
4272
4273 022030 104422 .CLKD1: MRCLK ;CLOCK MR REG WITH AN 11 AND A 1
4274 022032 005003 CLR R3 ;CLEAR WORK LOCATION
4275 022034 005767 157114 TST NOWOD ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
4276 022040 001005 BNE TSTEVB ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
4277 ;NOW BIT IS A 1 MWDB IS 0
4278 022042 005767 157102 1$: TST LSTOD ;TEST THE LAST ODD DATA BIT THAT WAS SENT
4279 022046 001002 BNE TSTEVB ;LAST ODD DATA BIT WAS A 1
4280 ;MWDB IS A 0
4281
4282 022050 052703 010000 2$: BIS #BIT12,R3 ;SET MWDB FOR LATER COMPARE WITH MR REG
4283 ;NOW TEST FOR EVEN BITS BEING SENT
4284
4285
4286 022054 005767 157072 TSTEVB: TST NOWEV ;TEST EVEN BIT NOW BEING TRANSFERRED
4287 ;FOR EITHER A 1 OR A 0
4288 022060 001005 BNE 1$ ;NOW BIT IS A 1
4289 022062 005767 157060 TST LSTEV ;WAS LAST EVEN DATA BIT A 0?
4290 022066 001002 BNE 1$ ;NO LAST EVEN DATA BIT WAS A 1
4291 022070 052703 040000 BIS #BIT14,R3 ;MWDT SHOULD BE SET
4292 022074 012701 123501 1$: MOV #123501,GOOD ;GET CORRECT ANS
4293 022100 050301 BIS R3,GOOD ;FOR MR REG
4294 022102 004767 001200 JSR PC,MRCAL ;DETERMINE STATE OF SB & LSR BITS
4295 022106 017700 157012 MOV @R5MR,BAD ;GET CONTENTS OF MR REG
4296 022112 020100 CMP GOOD,BAD ;IS MR REG CORRECT?
4297 022114 001002 BNE 2$ ;NO TYPE OUT MR REG
4298 022116 062716 000002 ADD #2,(SP) ;UPDATE RETURN ADDR FOR CORRECT ANS
4299 022122 000002 2$: RTI ;RETURN

```



```

4300 ;SECOND CLOCK ROUTINE WHICH WILL FINISH TRANSFERRING THE TWO DATA BITS
4301 ;THIS ROUTINE WILL CALCULATE WHAT MWDT AND MWDB SHOULD EQUAL IN THE
4302 ;MAINTENANCE REGISTER
4303
4304 022124 104422 .CLKD2: MRCLK ;CLOCK MR REG
4305 022126 005767 157022 TST NOWOD ;IS THE PRESENT DATA BIT A 1?
4306 022132 001403 BEQ 1$ ;NO IT IS A 0
4307 022134 052703 010000 BIS #BIT12,R3 ;SET MWDB FOR BIT BEING SENT IS A 1
4308 022140 000402 BR 2$
4309 022142 042703 010000 1$: BIC #BIT12,R3 ;CLEAR MWDB FOR PRESENT BIT IS A 0
4310 022146 005767 157000 2$: TST NOWEV ;IS PRESENT EVEN BIT A 1
4311 022152 001403 BEQ 3$ ;NO IT IS A 0
4312 022154 052703 040000 BIS #BIT14,R3 ;IT IS A 1 SET MWDT
4313 022160 000402 BR 4$
4314 022162 042703 040000 3$: BIC #BIT14,R3 ;PRESENT BIT IS A 0 CLEAR MWDT
4315 022166 012701 023501 4$: MOV #23501,GOOD ;GET CORRECT ANS
4316 022172 050301 BIS R3,GOOD ;FOR MR REG
4317 022174 004767 JSR PC,MRCAL ;DETERMINE STATE OF SB & LSR BITS
4318 022200 017700 156720 MOV #RSMR,BAD ;GET CONTENTS OF MR REG
4319 022204 020100 CMP GOOD,BAD ;IS MR REG CORRECT?
4320 022206 001002 BNE 5$ ;NO TYPEOUT ERROR
4321 022210 062716 000002 ADD #2,(SP) ;UPDATE RETURN ADDR FOR CORRECT ANS
4322 022214 000002 5$: RTI ;RETURN
4323
4324 ;TYPEOUT ROUTINE TO DETERMINE WHICH IC FAILED IN CRC TEST2
4325 ;AND TO TYPE IT OUT
4326
4327 022216 012767 022326 156770 CRCTYP: MOV #CRCTAB,WORK ;GET STARTING LOC OF IC TABLE
4328 022224 012767 000001 156766 MOV #1,WORK1 ;SETUP TO TEST FIRST CHIP
4329 022232 036767 156762 156732 1$: BIT WORK1,SAVEE ;WAS IT THIS BIT?
4330 022240 001006 BNE 2$ ;YES TYPE IT
4331 022242 062767 000006 156744 ADD #6,WORK ;NO INDEX TABLE POINTER
4332 022250 006167 156744 ROL WORK1 ;SETUP TO TEST NEXT CHIP
4333 022254 000766 BR 1$ ;NOW TES IT
4334 022256 004777 156732 2$: JSR PC,#WORK ;TYPE OUT CHIP
4335 022262 104402 022266 TYPE #2 ;.ASCIZ " IN THE CRC REG SHOULD BE SET"
4336 022324 000207 RTS PC

```

;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE

4337					
4338					
4339	022326	104402	022466	CRCTAB:	TYPE E302
4340	022332	000207			RTS PC
4341	022334	104402	022474		TYPE E305
4342	022340	000207			RTS PC
4343	022342	104402	022502		TYPE E307
4344	022346	000207			RTS PC
4345	022350	104402	022510		TYPE E3010
4346	022354	000207			RTS PC
4347	022356	104402	022517		TYPE E3012
4348	022362	000207			RTS PC
4349	022364	104402	022526		TYPE E3015
4350	022370	000207			RTS PC
4351	022372	104402	022535		TYPE E242
4352	022376	000207			RTS PC
4353	022400	104402	022543		TYPE E245
4354	022404	000207			RTS PC
4355	022406	104402	022551		TYPE E247
4356	022412	000207			RTS PC
4357	022414	104402	022557		TYPE E2410
4358	022420	000207			RTS PC
4359	022422	104402	022566		TYPE E2412
4360	022426	000207			RTS PC
4361	022430	104402	022575		TYPE E2415
4362	022434	000207			RTS PC
4363	022436	104402	022604		TYPE E192
4364	022442	000207			RTS PC
4365	022444	104402	022612		TYPE E197
4366	022450	000207			RTS PC
4367	022452	104402	022620		TYPE E1910
4368	022456	000207			RTS PC
4369	022460	104402	022627		TYPE E1915
4370	022464	000207			RTS PC

4371	022466	031505	026460	000062	E302:	.ASCIZ	"E30-2"
4372	022474	031505	026460	000065	E305:	.ASCIZ	"E30-5"
4373	022502	031505	026460	000067	E307:	.ASCIZ	"E30-7"
4374	022510	031505	026460	030061	E3010:	.ASCIZ	"E30-10"
4375	022516	000					
4376	022517	105	030063	030455	E3012:	.ASCIZ	"E30-12"
4377	022524	000062					
4378	022526	031505	026460	032461	E3015:	.ASCIZ	"E30-15"
4379	022534	000					
4380	022535	105	032062	031055	E242:	.ASCIZ	"E24-2"
4381	022542	000					
4382	022543	105	032062	032455	E245:	.ASCIZ	"E24-5"
4383	022550	000					
4384	022551	105	032062	033455	E247:	.ASCIZ	"E24-7"
4385	022556	000					
4386	022557	105	032062	030455	E2410:	.ASCIZ	"E24-10"
4387	022564	000060					
4388	022566	031105	026464	031061	E2412:	.ASCIZ	"E24-12"
4389	022574	000					
4390	022575	105	032062	030455	E2415:	.ASCIZ	"E24-15"
4391	022602	000065					
4392	022604	030505	026471	000062	E192:	.ASCIZ	"E19-2"
4393	022612	030505	026471	000067	E197:	.ASCIZ	"E19-7"
4394	022620	030505	026471	030061	E1910:	.ASCIZ	"E19-10"
4395	022626	000					
4396	022627	105	034461	030455	E1915:	.ASCIZ	"E19-15"
4397	022634	000065					

```

4398                                     ;GET TWO BITS OF DATA FROM INBUF
4399                                     ;FOR READING FROM DRIVE TO DETERMINE THE
4400                                     ;STATE OF MRDT AND MRDB IN THE MR REG.
4401
4402 022636 005767 156362                .RBIT: TST      WORK3      ;STARTING NEW WD?
4403 022642 001031                      BNE      3$          ;NO
4404 022644 062705 000002                ADD      #2,R5      ;UPDATE BUFFER WD
4405 022650 011504                      MOV      (R5),R4    ;GET DATA WD
4406 022652 005067 156330                CLR      CLKCNT     ;CLEAR CLOCK COUNTER AT START OF EACH WD
4407 022656 032767 000400 156302        BIT      #BIT8,ONCE ;ON CRC WD?
4408 022664 001035                      BNE      1$          ;YES
4409 022666 032767 000040 156250        BIT      #BITS,FLAG2 ;IN CRC TEST ???
4410 022674 001404                      BEQ      7$          ;NO
4411 022676 012767 000010 156320        MOV      #8.,WORK3
4412 022704 000410                      BR       3$
4413 022706 005067 156242                7$: CLR      NOWOD   ;LOAD EVEN & ODD BITS
4414 022712 005067 156234                CLR      NOWEV     ;WITH 0 FOR BITS 16 & 17 IN R504 DATA WORD
4415 022716 012767 000010 156300        6$: MOV      #8.,WORK3 ;8 LOOPS FOR REMAINING 16 BITS OF WORD
4416 022724 000002
4417 022726 005067 156222                3$: CLR      NOWOD   ;CLEAR PRESENT ODD BIT
4418 022732 006104                      ROL      R4        ;GET NEXT ODD DATA BIT
4419 022734 006167 156214                ROL      NOWOD     ;SAVE IT IN ODD BIT
4420 022740 005067 156206                CLR      NOWEV     ;CLEAR PRESENT EVEN BIT
4421 022744 006104                      ROL      R4        ;GET NEXT EVEN BIT
4422 022746 006167 156200                ROL      NOWEV     ;SAVE IT IN EVEN BIT
4423 022752 005367 156246                DEC      WORK3     ;KEEP COUNT OF BITS IN THE WORD
4424 022756 000002                      RTI
4425                                     ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
4426 022760 005067 156170                1$: CLR      NOWOD   ;GET BITS 17
4427 022764 006104                      ROL      R4        ;AND 16
4428 022766 006167 156162                ROL      NOWOD     ;FOR CRC WORD
4429 022772 005067 156154                CLR      NOWEV
4430 022776 006104                      ROL      R4
4431 023000 006167 156146                ROL      NOWEV
4432 023004 000744                      BR       6$
                                     ;CONTINUE

```


4433	023006	004767	000236		.CLKR1: JSR	PC, CALRTB	: CALCULATE TOP AND BOTTOM BITS FOR MR REG
4434	023012	012703	000011		MOV	#11, R3	: SETUP CLOCK BITS
4435	023016	056703	156172		CLOCK: BIS	WORK, R3	: SET TOP & BOTTOM BITS
4436	023022	010377	156076		MOV	R3, @RSMR	: SEND
4437	023026	042703	000010		BIC	#BIT3, R3	: CLOCK
4438	023032	010377	156066		MOV	R3, @RSMR	: PULSE
4439	023036	062757	000001	156132	ADD	#1, MCCNT+2	: INCREMENT
4440	023044	005567	156124		ADC	MCCNT	: CLOCK COUNT
4441	023050	012701	023601		MOV	#23601, GOOD	: CALCULATE CORRECT ANS FOR MR REG
4442	023054	032767	000004	156062	BIT	#BIT2, FLAG2	: WRITE CK TEST?
4443	023062	001402			BEQ	7\$: NO
4444	023064	052701	000100		BIS	#BIT6, GOOD	: YES SET RD IN MR REG
4445	023070	050301			7\$: BIS	R3, GOOD	
4446	023072	042701	000010		BIC	#BIT3, GOOD	: CLEAR MCLK
4447	023076	032767	000400	156062	BIT	#BIT8, ONCEE	: ON CRC WD?
4448	023104	001406			BEQ	5\$: NO
4449	023106	022767	000011	156066	CMP	#11, REPT	: SHOULD CRCW BE SET?
4450	023114	001402			BEQ	5\$: YES
4451	023116	042701	020000		BIC	#20000, GOOD	: CLEAR CRCW
4452	023122	032767	000001	156014	5\$: BIT	#BIT0, FLAG2	: SHOULD SDCLK BE SET?
4453	023130	001004			BNE	1\$: YES
4454	023132	052767	000001	156004	BIS	#BIT0, FLAG2	: NO
4455	023140	000405			BR	2\$: CONTINUE
4456	023142	052701	100000		1\$: BIS	#BIT15, GOOD	: SET IT
4457	023146	042767	000001	155770	BIC	#BIT0, FLAG2	: CLEAR FLAG FOR SDCLK FOR NEXT CLOCK PULSE
4458	023154	005367	156024		2\$: DEC	REPT1	: SHOULD SB SET?
4459	023160	001017			BNE	6\$: NO
4460	023162	012767	000022	156014	MOV	#18, REPT1	: RESET SB COUNTER
4461	023170	052701	004000		BIS	#BIT11, GOOD	: SET SB
4462	023174	032767	000400	155764	3\$: BIT	#BIT8, ONCEE	: ON CRC WD?
4463	023202	001406			BEQ	6\$: NO
4464	023204	022767	000022	155772	CMP	#22, REPT1	: SHOULD SB AND CRCW BE SET ?
4465	023212	001002			BNE	6\$: NO
4466	023214	052701	020000		BIS	#20000, GOOD	: SET SB AND CRCW
4467	023220	017700	155700		6\$: MOV	@RSMR, BAD	: GET MR REG
4468	023224	020100			CMP	GOOD, BAD	: IS RSMR CORRECT?
4469	023226	001002			BNE	4\$: NO
4470	023230	062716	000002		ADD	#2, (SP)	: YES
4471	023234	000002			4\$: RTI		: RETURN
4472							
4473	023236	004767	000006		.CLKR2: JSR	PC, CALRTB	
4474	023242	012703	050011		MOV	#50011, R3	
4475	023246	000663			BR	CLOCK	

```

4476 ;CALCULATE THE STATE OF MRDT AND MRDB FROM CURRENT INPUT BITS.
4477 ;LOCATION WORK CONTAINS CORRECT DATA FOR MRDT AND MRDB
4478 023250 005067 155740 CALRTB: CLR WORK ;CLEAR WORK LOCATION
4479 023254 005767 155674 TST NOWOD ;IS CURRENT ODD BIT A 0?
4480 023260 001403 BEQ 1$ ;YES
4481 023262 052767 000004 155724 BIS #BIT2,WORK ;NO SET MRDB
4482 023270 005767 155656 1$: TST NOWEV ;IS CURRENT EVEN BIT A 0?
4483 023274 001403 BEQ 2$ ;YES
4484 023276 052767 000040 155710 BIS #BITS,WORK ;NO SET MRDT
4485 023304 000207 2$: RTS PC ;RETURN
4486
4487 ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
4488 ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR
4489
4490 023306 005267 155674 MRCAL: INC CLKCNT ;ADD ONE TO CLOCK COUNT OF WORD
4491 023312 032767 000200 155646 BIT #BIT7,ONCEE ;TRANSFERRING LAST WORD?
4492 023320 001026 BNE LSTWD ;YES
4493 023322 032767 000400 155636 BIT #BIT8,ONCEE ;TRANSFERRING CRC WORD?
4494 023330 001040 BNE CRCWD ;YES
4495 023332 022767 000010 155646 CMP #8.,CLKCNT ;CLOCK COUNT 8 OR GREATER?
4496 023340 101401 BLOS 1$ ;YES
4497 023342 000414 BR 2$ ;GET OUT
4498 023344 022767 000021 155634 1$: CMP #17.,CLKCNT ;CLOCK COUNT 17 OR GREATER?
4499 023352 101410 BLOS 2$ ;YES GET OUT
4500
4501 023354 052701 004000 BIS #BIT11,GOOD ;SET SB BIT
4502 023360 022767 000017 155620 CMP #15.,CLKCNT ;SHOULD LSR BE CLEARED
4503 023366 001002 BNE 2$ ;NO
4504 023370 042701 002000 BIC #BIT10,GOOD ;CLEAR LSR
4505 023374 000207 2$: RTS PC ;RETURN
4506
4507 ;CALCULATE MR FOR LAST DATA WORD
4508 023376 022767 000016 155602 LSTWD: CMP #14.,CLKCNT ;IS THIS CLOCK 14 OR LESS?
4509 023404 103011 BHIS 2$ ;YES GETOUT
4510 023406 022767 000017 155572 CMP #15.,CLKCNT ;IS THIS CLOCK 15?
4511 023414 001003 BNE 1$ ;NO
4512 023416 042701 002000 BIC #BIT10,GOOD ;YES CLEAR LSR
4513 023422 000402 BR 2$ ;GET OUT
4514 023424 042701 020000 1$: BIC #BIT13,GOOD ;CLEAR CRCW BIT
4515 023430 000207 2$: RTS PC
4516
4517 ;CALCULATE MR FOR CRC WORD
4518
4519 023432 042701 020000 CRCWD: BIC #BIT13,GOOD ;CLEAR CRCW BIT
4520 023436 022767 000017 155542 CMP #17,CLKCNT ;IS THIS CLOCK 17?
4521 023444 001002 BNE 1$ ;NO
4522 023446 042701 002000 BIC #BIT10,GOOD ;CLEAR LSR BIT
4523 023452 000207 1$: RTS PC ;RETURN

```



```

4524
4525 ;GENERATE A CRC WORD FROM THE DATA BUFFER
4526 ;AND LEAVE THE CRC WORD IN "WORK" LOCATION
4527 ;EXIT ROUTINE WITH RTS PC
4528
4529 023454 012767 000200 155520 GENCRC: MOV #128.,REPT ;128 WORDS PER SECTOR
4530 023462 032767 000040 155454 BIT #BITS,FLAG2 ;IN CRC TEST?
4531 023470 001403 BEQ 13$ ;NO
4532 023472 012767 000220 155502 MOV #144.,REPT ;YES
4533 023500 012705 026572 13$: MOV #INBUF,R5 ;GET STARTING ADDR OF OUTPUT BUFFER
4534 023504 011504 MOV (R5),R4 ;GET DATA WD
4535 023506 005067 155504 CLR WORK0 ;CLEAR WORK LOCATION
4536
4537 ;INBIT CONTAINS PRESENT INPUT BIT
4538 ;WK15 = BIT15 OF CRC AT TIME T
4539 ;WORK0 = CRC AT TIME T + DURING FINAL MANIPULATION
4540 ;WORK = BITS FROM SAVED CRC WORD (WCRC)
4541
4542 023512 012767 000022 155464 1$: MOV #18.,REPT1 ;GET 18 BITS PER WD
4543 023520 032767 000040 155416 BIT #BITS,FLAG2 ;IN CRC TEST?
4544 023526 001403 BEQ 2$ ;NO
4545 023530 012767 000020 155446 MOV #16.,REPT1 ;YES
4546 023536 016767 155454 155434 2$: MOV WORK0,WCRC ;SAVE CURRENT CRC WD
4547 023544 005067 155442 CLR WK15 ;CLEAR BIT 15 FROM CRC AT T 1
4548 023550 000241 CLC ;CLEAR CARY
4549 023552 006167 155440 ROL WORK0 ;SHIFT CRC WD LEFT
4550 023556 006167 155430 ROL WK15 ;CONTAINS BIT 15 OF CRC
4551 023562 032767 000040 155354 BIT #BITS,FLAG2 ;IN CRC TEST?
4552 023570 001004 BNE 12$ ;YES
4553 023572 022767 000021 155404 CMP #17.,REPT1 ;DONE BITS 16 AND 17 YET?
4554 023600 101406 BLOS 3$ ;NO
4555 023602 005067 155402 12$: CLR INBIT ;CLEAR WORK LOC
4556 023606 006104 ROL R4 ;PUT DATA BIT FROM BUFFER
4557 023610 006167 155374 ROL INBIT ;IN WORK1 LOC
4558 023614 000402 BR 4$
4559 023616 005067 155366 3$: CLR INBIT ;FOR BITS 16 AND 17
4560 023622 016767 155364 155364 4$: MOV WK15,WORK ;GET BIT 15 OF CRC
4561 023630 004767 000220 155354 5$: JSR PC,XXOR ;XOR BIT15 WITH INPUT BIT
4562 023634 042767 000001 155354 BIC #BIT0,WORK0
4563 023642 005767 155342 TST INBIT ;TEST RESULT OF XOR
4564 023646 001403 BEQ 6$
4565 023650 052767 000001 155340 BIS #BIT0,WORK0
4566 023656 016767 155326 155272 6$: MOV INBIT,RS0 ;SAVE XOR RESULT OF BIT 0 AND INPUT

```

```

4567          ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CALCULATE
4568          ;NEW B2 FOR WORK0
4569
4570 023664 005067 155324          CLR      WORK
4571 023670 032767 000002 155302  BIT      #BIT1,WCRC
4572 023676 001403          BEQ      7$
4573 023700 052767 000001 155306  BIS      #BIT0,WORK
4574 023706 016767 155244 155274 7$:  MOV      R50,INBIT
4575 023714 004767 000134          JSR      PC,XXOR
4576 023720 042767 000004 155270  BIC      #BIT2,WORK0
4577 023726 005767 155256          TST      INBIT          ;TEST RESULT OF XOR
4578 023732 001403          BEQ      8$
4579 023734 052767 000004 155254  BIS      #BIT2,WORK0
4580
4581          ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
4582
4583 023742 005067 155246          8$:  CLR      WORK
4584 023746 032767 040000 155224  BIT      #BIT14,WCRC
4585 023754 001403          BEQ      9$
4586 023756 052767 000001 155230  BIS      #BIT0,WORK
4587 023764 016767 155166 155216 9$:  MOV      R50,INBIT
4588 023772 004767 000056          JSR      PC,XXOR
4589 023776 042767 100000 155212  BIC      #BIT15,WORK0
4590 024004 005767 155200          TST      INBIT          ;TEST RESULT OF XOR
4591 024010 001403          BEQ      10$
4592 024012 052767 100000 155176  BIS      #BIT15,WORK0
4593 024020 005367 155160          10$: DEC      REPT1          ;DONE WITH WD
4594 024024 001244          BNE      2$              ;NO
4595 024026 005367 155150          DEC      REPT          ;DONE WITH SECTOR?
4596 024032 001404          BEQ      11$           ;YES
4597 024034 062705 000002          ADD      #2,R5          ;GET NEXT WD
4598 024040 011504          MOV      (R5),R4       ;GET DATA WD
4599 024042 000623          BR       1$
4600 024044 016767 155146 155142 11$: MOV      WORK0,WORK    ;SAVE CRC WORD IN WORK
4601 024052 000207          RTS      PC           ;EXIT
4602
4603          ;XOR SUBROUTINE
4604
4605 024054 016703 155134          XXOR:  MOV      WORK,R3
4606 024060 046703 155124          BIC      INBIT,R3
4607 024064 046767 155124 155116  BIC      WORK,INBIT
4608 024072 050367 155112          BIS      R3,INBIT
4609 024076 000207          RTS      PC

```


4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620 024100 010446
4621 024102 010546
4622 024104 017605 000004
4623 024110 032705 177400
4624 024114 001002
4625 024116 016605 000004
4626 024122 105715
4627 024124 001423
4628 024126 122715 000012
4629 024132 001012
4630 024134 116704 154655
4631 024140 116777 154650 154652
4632 024146 105777 154644
4633 024152 100375
4634 024154 005304
4635 024156 001370
4636 024160 112577 154634
4637 024164 105777 154626
4638 024170 100375
4639 024172 000753
4640 024174 017646 000004
4641 024200 062766 000002 000006
4642 024206 022666 000004
4643 024212 001006
4644 024214 062705 000002
4645 024220 042705 000001
4646 024224 010566 000004
4647 024230 012605
4648 024232 012604
4649 024234 000002

.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 INLINE ASCII. THE FILLER CHARACTER WHICH IS
; TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
; IS IN FILCHR+1.
.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 TYPEN?
BNE IS ;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 IS ;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(6)+,4(6) ;IS IT .+2?
BNE 3\$;NO
ADD #2,R5 ;ADD 2 TO THE ADDRESS
BIC #1,R5 ;BACK UP TO AN EVEN BYTE
MOV R5,4(6) ;RESTORE ADDRESS
3\$: MOV 6(6)+,R5 ;RESTORE R5
MOV 6(6)+,R4 ;RESTORE R4
RTI ;RETURN

```

4650          .SBTTL          $SCOPE - SCOPE LOOP HANDLER
4651
4652          ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
4653          ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
4654          ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
4655          ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
4656
4657 024236 032737 000400 177570 .SCOPE: BIT      #SW8, @SWR      ;LOOP ON SPEC. TEST?
4658 024244 001404          BEQ      1$          ;NO LOOP ON SPEC. TEST
4659 024246 123767 177570 154524 CMPB    @SWR, ICNT    ;ON RIGHT TEST? *SW7-0*
4660 024254 001453          BEQ      .OVER      ;NOT RIGHT TEST
4661 024256 032737 040000 177570 1$: BIT      #SW14, @SWR     ;LOOP ON TEST?
4662 024264 001045          BNE      .KIT      ;LOOP ON TEST IS SET
4663 024266 000416          BR       3$          ;SKIP - NOP FOR XOR TESTER
4664 024270 013746 000004          MOV     @#4, -(6)    ;PUSH @#4 ON STACK
4665 024274 012737 024314 000004 MOV     #4$, @#4     ;SET FOR TIMEOUT
4666 024302 005737 177060          TST    @#177060    ;ERROR ON XOR?
4667 024306 012637 000004          MOV     (6)+, @#4   ;POP STACK INTO @#4
4668 024312 000422          BR       .SVLAD     ;NO ERROR - GO TO NEXT TEST
4669 024314 022626          4$: CMP     (6)+, (6)+ ;CLEAR STACK
4670 024316 012637 000004          MOV     (6)+, @#4   ;POP STACK INTO @#4
4671 024322 000426          BR       .KIT      ;ERROR - LOOP ON TEST
4672 024324 032737 004000 177570 3$: BIT      #SW11, @SWR     ;KILL ITERATIONS
4673 024332 001012          BNE      .SVLAD     ;YES - KILL ITERATIONS
4674 024334 105767 154441          TSTB   ICNT+1      ;FIRST ONE?
4675 024340 001404          BEQ     2$          ;BRANCH IF FIRST
4676 024342 126767 000060 154431 CMPB    TIMES, ICNT+1 ;DONE?
4677 024350 003013          BGT     .KIT      ;BRANCH IF NOT
4678 024352 112767 000001 154421 2$: MOVB   #1, ICNT+1    ;FIRST ITERATION
4679 024360 105267 154414          .SVLAD: INCB   ICNT    ;COUNT TEST NUMBERS
4680 024364 011667 154420          MOV     (6), LAD    ;SAVE LOOP ADDRESS
4681 024370 016737 154404 177570 MOV     ICNT, @#DISPLAY ;DISPLAY TEST NO. AND ITERATION COUNT
4682 024376 000002          RTI          ;RETURN
4683
4684 024400 105267 154375          .KIT: INCB   ICNT+1    ;INC THE ITERATION COUNT
4685 024404 016737 154370 177570 .OVER: MOV     ICNT, @#DISPLAY ;SET UP DISPLAY
4686 024412 005767 154372          TST    LAD         ;FIRST ONE?
4687 024416 001760          BEQ     .SVLAD     ;YES
4688 024420 016716 154364          MOV     LAD, (6)   ;FUDGE RETURN ADDRESS
4689 024424 000002          RTI          ;FIXES PS
4690
4691 024426 000001          TIMES: 1          ;RUN 1 TIMES

```



```

4692          .SBTTL          SHLT - HLT ROUTINE (ERROR TYPEOUT)
4693
4694          ; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
4695          ; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
4696          ; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
4697          ; "HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
4698          ; (HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITONAL TYPEOUTS.
4699
4700 024430 032737 002000 177570 .HLT: BIT      #SW10,@#SWR      ; BELL ON ERROR?
4701 024436 001402          BEQ      1$              ; NO - SKIP
4702 024440 104402          TYPE     .BELL           ; RING BELL
4703 024444 005267 154332 1$: INC      ERRORS        ; COUNT THE NUMBER OF ERRORS
4704 024450 032737 020000 177570 BIT      #SW13,@#SWR      ; SKIP TYPEOUT IF SET
4705 024456 001025          BNE      2$              ; SKIP TYPEOUTS
4706 024460 104402 024464          TYPE     .+2          ; .ASCIZ <15><12>
4707 024470 011667 154316          MOV      (6),HLTADR      ; PUT ADDRESS OF INSTRUCTION ON STACK
4708 024474 162767 000002 154310 SUB      #2,HLTADR      ; FUDGE ADDRESS
4709 024502 117767 154304 000054 MOVB     @HLTADR,.HLTCT ; GET HLT ARGUMENT
4710 024510 016746 154276          MOV      HLTADR,-(6)   ; PUT HLTADR ON STACK
4711 024514 104404          TYPEO     TYPE          ; TYPE STACK IN OCTAL
4712 024516 104402 024522          TYPE     .+2          ; .ASCIZ " "
4713 024526 004767 001140          JSR      PC,RSREG      ; GO TO USER ERROR ROUTINE
4714 024532 005737 177570 2$: TST      @#SWR          ; HALT ON ERROR
4715 024536 100001          BPL      .+4          ; SKIP IF CONTINUE
4716 024540 000000          HALT          ; HALT ON ERROR!
4717 024542 032737 001000 177570 BIT      #SW9,@#SWR      ; CHECK FOR INHIBIT LOOP ON ERROR
4718 024550 001003          BNE      3$              ; SKIP IF LOOP ON ERROR
4719 024552 105067 154223          CLRB     ICNT+1      ; CLEAR ITERATION COUNT
4720 024556 000002          RTI          ; RETURN
4721 024560 000167 177614 3$: JMP      .KIT          ; LOOP ON TEST UNTIL NO ERRORS
4722
4723 024564 000000          .HLTCT: 0          ; HLT ARGUMENT

```

```

4724          .SBTTL          SOCTAL - OCTAL TYPEOUT ROUTINE
4725
4726          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
4727          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
4728          ;16 BITS. IT IS CALLED VIA THE TYOCT, TYPBIT, OR TYPOCS MACRO'S.
4729
4730 024566 012767 170101 000160 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
4731 024574 000411                BR          .PTIT          ;NOW TYPE IT IN BIT FORM
4732 024576 112767 000001 000150 .TYPEO: MOVB     #1,.PR          ;SET ZERO FILL SWITCH
4733 024604 000402                BR          .+6          ;SKIP
4734 024606 005067 000142                .TYPES: CLR      .PR          ;SUPPRESS LEADING ZERO'S
4735 024612 112767 177772 000135                MOVB     #-6,.PR+1      ;SET COUNT
4736 024620
4737 024620 010446                MOV      R4,-(6)      ;PUSH R4 ON STACK
4738 024622 010546                MOV      R5,-(6)      ;PUSH R5 ON STACK
4739 024624 016605 000010                MOV      10(6),R5     ;GET THE DATA
4740 024630 012704 024756                MOV      #.PR+2,R4    ;SET POINTER TO FIRST ASCII CHAR.
4741 024634 105014                CLRB     (4)          ;CLEAR FIRST BYTE
4742 024636 000411                BR          .PRF       ;ROTATE FIRST BIT
4743 024640 105014                .PRL:  CLRB     (4)    ;CLEAR BYTE OF CHARACTER
4744 024642 032767 000100 000104                BIT      #100,.PR     ;BIT TYPING MODE?
4745 024650 001004                BNE      .PRF         ;YES - SKIP 2 ROTATES
4746 024652 006105                ROL      R5           ;ROTATE BIT INTO C
4747 024654 106114                ROLB     (4)          ;PACK IT
4748 024656 006105                ROL      R5           ;ROTATE BIT INTO C
4749 024660 106114                ROLB     (4)          ;PACK IT
4750 024662 006105                .PRF:  ROL      R5     ;ROTATE BIT INTO C
4751 024664 106114                ROLB     (4)          ;PACK IT
4752 024666 105714                TSTB     (4)          ;IS IT ZERO?
4753 024670 001402                BEQ      .+6          ;SKIP INC
4754 024672 105267 000056                INCB     .PR          ;SET FILL SWITCH
4755 024676 105767 000052                TSTB     .PR          ;CHECK FILL SWITCH
4756 024702 001402                BEQ      .+6          ;SKIP BITSET
4757 024704 152724 000060                BISB     #'0,(4)+    ;MAKE INTO ASCII CHAR
4758 024710 105267 000041                INCB     .PR+1       ;INC COUNT
4759 024714 001351                BNE      .PRL        ;REPEAT
4760 024716 022704 024756                CMP      #.PR+2,R4    ;EMPTY BUFFER?
4761 024722 001002                BNE      .+6          ;SKIP IF NOT
4762 024724 112724 000060                MOVB     #'0,(4)+    ;LOAD 1 ZERO
4763 024730 105014                CLRB     (4)          ;NULL TERMINATOR
4764 024732 104402 024756                TYPE     .PR+2       ;TYPE IT
4765 024736 012605                MOV      (6)+,R5      ;POP STACK INTO R5
4766 024740 012604                MOV      (6)+,R4      ;POP STACK INTO R4
4767 024742 016666 000002 000004                MOV      2(6),4(6)   ;GET RID OF
4768 024750 012616                MOV      (6)+,(6)    ;DATA WORD
4769 024752 000002                RTI
4770
4771 024754 000012                .PR:   .BLKW 12      ;COUNT, SWITCH, AND OUTPUT BUFFER

```



```

4772          .SBTTL          $POWER - POWER DOWN AND UP ROUTINES
4773
4774          ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
4775          ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
4776          ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
4777          ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
4778          ;THE PROGRAM WILL HALT AT '.ILLUP'.
4779
4780 025000 012777 025126 000126 .POWER: MOV      #.ILLUP, @.PUVEC ;SET FOR FAST UP
4781 025006 012777 000340 000122      MOV      #340, @.PUVECS+2 ;PRIO:7
4782 025014 010046          MOV      R0, -(6) ;PUSH R0 ON STACK
4783 025016 010146          MOV      R1, -(6) ;PUSH R1 ON STACK
4784 025020 010246          MOV      R2, -(6) ;PUSH R2 ON STACK
4785 025022 010346          MOV      R3, -(6) ;PUSH R3 ON STACK
4786 025024 010446          MOV      R4, -(6) ;PUSH R4 ON STACK
4787 025026 010546          MOV      R5, -(6) ;PUSH R5 ON STACK
4788 025030 010667 000076          MOV      SP, .SAVR6 ;SAVE SP
4789 025034 012777 025044 000072      MOV      #.POWUP, @.PUVEC ;SET UP VECTOR
4790 025042 000000          HALT      ;WAIT FOR PF
4791
4792 025044 016706 000062          .POWUP: MOV      .SAVR6, SP ;GET SP
4793 025050 005001          CLR      R1 ;WAIT LOOP FOR THE TTY
4794 025052 005201          1$: INC     R1 ;WAIT FOR THE INC
4795 025054 001376          BNE     1$ ;OF WORD
4796 025056 012605          MOV      (6)+, R5 ;POP STACK INTO R5
4797 025060 012604          MOV      (6)+, R4 ;POP STACK INTO R4
4798 025062 012603          MOV      (6)+, R3 ;POP STACK INTO R3
4799 025064 012602          MOV      (6)+, R2 ;POP STACK INTO R2
4800 025066 012601          MOV      (6)+, R1 ;POP STACK INTO R1
4801 025070 012600          MOV      (6)+, R0 ;POP STACK INTO R0
4802 025072 012737 025000 000024      MOV      #.POWER, @#24 ;SET UP THE POWER DOWN VECTOR
4803 025100 012737 000340 000026      MOV      #340, @#26 ;PRIO:7
4804 025106 104402 025112          TYPE   +2 ;.ASCIZ <15><12>"POWER"
4805 025122 000167 174066          JMP     MULSYS ;JMP TO USER ADDRESS
4806
4807 025126 000000          .ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
4808 025130 000776          BR      .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
4809
4810 025132 000000          .SAVR6: 0 ;PUT THE SP HERE
4811 025134 000024 000026          .PUVEC: 24, 26 ;POWER UP VECTOR

```

```

4812          .SBTTL          SRDOCT - OCTAL INPUT ROUTINE
4813
4814          ; THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
4815          ; IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
4816
4817 025140 011646          .RDOCT: MOV      4(6), - (6)          ; MOVE THE PC
4818 025142 016666 000004 000002 MOV      4(6), 2(6)          ; MOVE THE PS
4819 025150 010146          MOV      R1, - (6)          ; PUSH R1 ON STACK
4820 025152 010246          MOV      R2, - (6)          ; PUSH R2 ON STACK
4821 025154 010346          MOV      R3, - (6)          ; PUSH R3 ON STACK
4822 025156 104412          4$: RDLIN          ; READ A LINE INTO INPUT
4823 025160 005001          CLR      R1          ; INIT DATA WORD
4824 025162 012703 025362 MOV      #INPUT, R3          ; INIT POINTER
4825 025166 112302          1$: MOVB   (3)+, R2          ; GET A BYTE
4826 025170 001417          BEQ     2$          ; GET OUT IF ZERO
4827 025172 122702 000060 CMPB   #'0, R2          ; CHECK FOR 0 OR GREATER
4828 025176 003022          BGT     3$          ; ERROR - LESS THAN 0
4829 025200 122702 000067 CMPB   #'7, R2          ; CHECK FOR 7 OR LESS
4830 025204 002417          BLT     3$          ; ERROR - GREATER THAN 7
4831 025206 006002          ROR     R2          ; GET
4832 025210 006002          ROR     R2          ; INTO
4833 025212 006002          ROR     R2          ; POSITION
4834 025214 006101          ROL     R1          ; FIRST BIT
4835 025216 006102          ROL     R2          ; GET
4836 025220 006101          ROL     R1          ; SECOND BIT
4837 025222 005102          ROL     R2          ; GET
4838 025224 006101          ROL     R1          ; THIRD BIT
4839 025226 000757          BR      1$          ; LOOP
4840 025230 010166 000012 2$: MOV      R1, 12(6)          ; SAVE THE RESULT
4841 025234 012603          MOV     (6)+, R3          ; POP STACK INTO R3
4842 025236 012602          MOV     (6)+, R2          ; POP STACK INTO R2
4843 025240 012601          MOV     (6)+, R1          ; POP STACK INTO R1
4844 025242 000002          RTI
4845
4846 025244          3$:
4847 025244 104402 025250 TYPE   4$+2          ; .ASCIZ "'<15><12>"
4848 025254 000740          BR      4$          ; TRY AGAIN

```



```

4849
4850
4851
4852
4853
4854
4855
4856 025256 010546
4857 025260 012705 025362
4858 025264 022705 025402
4859 025270 001412
4860 025272 105737 177560
4861 025276 100375
4862 025300 113715 177562
4863 025304 142715 000200
4864 025310 122715 000177
4865 025314 001005
4866 025316
4867 025316 104402 025322
4868 025326 000754
4869 025330 111527 000000
4870 025334 104402 025332
4871 025340 122725 000015
4872 025344 001347
4873 025346 105065 177777
4874 025352 104402 000012
4875 025356 012605
4876 025360 000002
4877
4878 025362 000020
    
```

```

.SBTTL          SRDLIN - TTY INPUT ROUTINE

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

.RDLIN: MOV      R5, -(6)          ; SAVE R5
15:     MOV      #INPUT, R5       ; GET ADDRESS
25:     CMP      #INPUT+16., R5    ; BUFFER FULL?
        BEQ      45               ; YES - TYPE "?"
        TSTB    @177560           ; WAIT FOR
        BPL     -4                ; A CHARACTER
        MOVB    @177562, (5)      ; GET CHARACTER
        BICB    #200, (5)         ; GET RID OF JUNK
        CMPB    #177, (5)        ; IS IT A RUBOUT
        BNE     35               ; SKIP IF NOT

45:     TYPE    .+2               ; .ASCIZ "?"(15)<(12)
        BR     15                ; ZAP THE BUFFER AND LOOP
35:     MOVB    (5), #0           ; SET UP FOR TYPING
        TYPE    , 35+2           ; ECHO IT
        CMPB    #15, (5)+        ; CHECK FOR RETURN
        BNE     25               ; LOOP IF NOT RETURN
        CLRB   -1(5)             ; ZAP RETURN (THE 15)
        TYPE    12               ; TYPE A LINE FEED
        MOV     (6)+, R5         ; RESTORE R5
        RTI                      ; RETURN

INPUT:  .BLKB  16.              ; TTY INPUT AREA
    
```

```

4879
4880
4881
4882
4883
4884
4885
4886 025402 011646
4887 025404 162716 000002
4888 025410 017616 000000
4889 025414 062716 121022
4890 025420 013607
4891
4892 025422 024236
4893 025424 024100
4894 025426 024576
4895 025430 024606
4896 025432 025140
4897 025434 025256
4898 025436 025474
4899 025440 025522
4900 025442 021536
4901 025444 021506
4902 025446 021500
4903 025450 021602
4904 025452 021564
4905 025454 021630
4906 025456 022030
4907 025460 022124
4908 025462 023006
4909 025464 023236
4910 025466 022636
4911 025470 025614
4912 025472 025654

```

.SBTTL \$TRAP - TRAP HANDLER

```

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

```

```

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

```

```

.SCOPE = TRAP+0 (104400)
.TYPE = TRAP+2 (104402)
.TYPE0 = TRAP+4 (104404)
.TYPES = TRAP+6 (104406)
.RDOCT = TRAP+10 (104410)
.RDLIN = TRAP+12 (104412)
.CLRDK = TRAP+14 (104414)
.MRDMD = TRAP+16 (104416)
.MRCK = TRAP+20 (104420)
.MRCLK = TRAP+22 (104422)
.MRINT = TRAP+24 (104424)
.DSCK = TRAP+26 (104426)
.MRIND = TRAP+30 (104430)
.XBIT = TRAP+32 (104432)
.CLKD1 = TRAP+34 (104434)
.CLKD2 = TRAP+36 (104436)
.CLKR1 = TRAP+40 (104440)
.CLKR2 = TRAP+42 (104442)
.RBIT = TRAP+44 (104444)
.GETSP = TRAP+46 (104446)
.SPASS = TRAP+50 (104450)

```



```

4913                                     ;CLEAR ALL DISK REGISTERS
4914 025474 012777 000040 153400 .CLRDK: MOV #40,DRSCS2 ;CLEAR ALL DSK REG
4915 025502 016777 153452 153372          MOV UNNUM,DRSCS2 ;GET UNIT NUMBER
4916 025510 005067 153460          CLR MCCNT ;CLEAR MAINT CLOCK COUNT
4917 025514 005067 153456          CLR MCCNT+2
4918 025520 000002          RTI
4919
4920 025522 012777 000001 153374 .MRDMD: MOV #1,DRSMR ;PUT DRIVE INTO MAINT MODE
4921 025530 000002          RTI
4922
4923 025532 005067 153456          WAITRY: CLR WORK ;CLEAR COUNTER
4924 025536 105777 153336          1$: TSTB DRSCS1 ;TEST READY
4925 025542 100406          BMI 2$ ;OK CONT
4926 025544 005267 153444          INC WORK ;UPDATE COUNTER
4927 025550 005767 153440          TST WORK ;DONE YET?
4928 025554 001403          BEQ 3$ ;READY DID NOT COME UP
4929 025556 000767          BR 1$ ;CONTINUE WAITING
4930 025560 062716 000002          2$: ADD #2,(SP) ;UPDATE RETURN PC
4931 025564 000207          3$: RTS PC ;RETURN
4932
4933                                     ;ROUTINE TO SHIFT COMPLETE DATA TABLE ONE BIT
4934                                     ;TO THE LEFT. CARRIES BIT 15 OF ONE WORD TO BIT 0 OF THE NEXT WORD
4935
4936 025566 012702 026572          MDATA: MOV #INBUF,R2 ;GET LEFT ADDRESS OF
4937 025572 062702 000442          ADD #442,R2 ;DATA TABLE
4938 025576 012703 000220          MOV #220,R3 ;SETUP COUNTER FOR 200 WORDS
4939 025602 000241          CLC ;CLEAR CARRY
4940 025604 006142          1$: ROL -(R2) ;SHIFT DATA PATTERN
4941 025606 005303          DEC R3 ;DO ALL
4942 025610 001375          BNE 1$ ;WORDS
4943 025612 000207          RTS PC
4944
4945                                     ;THIS ROUTINE CLOCKS MR REG TO GET A SECTOR PULSE WHICH
4946                                     ;CLEARS OUT REGS. AND COUNTERS
4947
4948 025614 012767 002001 153360 .GETSP: MOV #1025.,REPT ;SETUP COUNTER
4949 025622 104430          MRIND ;SEND INDEX PULSE TO MR REG
4950 025624 104422          1$: MRCLK ;CLOCK MR
4951 025626 005367 153350          DEC REPT ;TO REACH
4952 025632 001374          BNE 1$ ;SECTOR PULSE
4953 025634 032777 000400 153262          BIT #400,DRSMR ;DID SECTOR PULSE SET?????
4954 025642 001401          BEQ 2$ ;YES
4955 025644 000002          RTI ;NO REPORT ERROR
4956 025646 062716 000002          2$: ADD #2,(SP) ;UPDATE RETURN ADDR
4957 025652 000002          RTI
4958
4959 025654 104422          .SPASS: MRCLK ;CLOCK PAST SECTOR PULSE
4960 025656 104422          MRCLK
4961 025660 005067 153310          CLR MCCNT ;RESET MAINT CLOCK COUNTERS
4962 025664 005067 153306          CLR MCCNT+2
4963 025670 000002          RTI

```

F10

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
STRAP - TRAP HANDLER

MACY11 27(732) 04-OCT-76 13:11 PAGE 123

```

;ERROR TYPTXTOUT ROUTINE
4964
4965
4966 025672 005767 176666      RSREG:  TST      .HLTCT      ; SHOULD WE TYPTXT GOOD AND BAD
4967 025676 001022                BNE      BS                ; NO
4968 025700 104402 025704      TYPE    .+2                ; .ASCIZ " BAD="
4969 025712 010046                MOV     BAD,-(6)           ; PUT BAD ON STACK
4970 025714 104404                TYPE0   ; TYPE STACK IN OCTAL
4971 025716 104402 025722      TYPE    .+2                ; .ASCIZ " GOOD="
4972 025732 010146                MOV     GOOD,-(6)         ; PUT GOOD ON STACK
4973 025734 104404                TYPE0   ; TYPE STACK IN OCTAL
4974 025736 000402                BR      BS                ; TYPEOUT REGISTERS
4975 025740 000167 000432      JMP     PTDONE            ; GET OUT
4976 025744
4977 025744 104402 025750      BS:    TYPE    .+2                ; .ASCIZ " CS1="
4978 025756 017746 153116      MOV     @RSCS1,-(6)       ; PUT @RSCS1 ON STACK
4979 025762 104404                TYPE0   ; TYPE STACK IN OCTAL
4980 025764
4981 025764 104402 025770      1S:    TYPE    .+2                ; .ASCIZ " ER="
4982 025776 017746 153112      MOV     @RSER,-(6)       ; PUT @RSER ON STACK
4983 026002 104404                TYPE0   ; TYPE STACK IN OCTAL
4984 026004
4985 026004 104402 026010      2S:    TYPE    .+2                ; .ASCIZ " CS2="
4986 026016 017746 153060      MOV     @RSCS2,-(6)       ; PUT @RSCS2 ON STACK
4987 026022 104404                TYPE0   ; TYPE STACK IN OCTAL
4988 026024 032767 000200 176532      BIT     #200,.HLTCT      ; TYPTXT SECOND SET ?
4989 026032 001076                BNE     SEEC              ; YES
4990 026034 032767 000100 176522      BIT     #AS,.HLTCT       ; TYPTXT ER ?
4991 026042 001410                BEQ     3S                ; NO
4992 026044 104402 026050      TYPE    .+2                ; .ASCIZ " AS="
4993 026056 017746 153034      MOV     @RSAS,-(6)       ; PUT @RSAS ON STACK
4994 026062 104404                TYPE0   ; TYPE STACK IN OCTAL
4995 026064 032767 000020 176472 3S:    BIT     #BA,.HLTCT       ; TYPTXT BUS ADDRESS
4996 026072 001410                BEQ     4S                ; NO
4997 026074 104402 026100      TYPE    .+2                ; .ASCIZ " BA="
4998 026106 017746 152774      MOV     @RSBA,-(6)       ; PUT @RSBA ON STACK
4999 026112 104404                TYPE0   ; TYPE STACK IN OCTAL
5000 026114 032767 000004 176442 4S:    BIT     #DA,.HLTCT       ; TYPTXT DA ?
5001 026122 001410                BEQ     5S                ; NO
5002 026124 104402 026130      TYPE    .+2                ; .ASCIZ " DA="
5003 026136 017746 152746      MOV     @RSDA,-(6)       ; PUT @RSDA ON STACK
5004 026142 104404                TYPE0   ; TYPE STACK IN OCTAL
5005 026144 032767 000010 176412 5S:    BIT     #WC,.HLTCT       ; TYPTXT WC?
5006 026152 001410                BEQ     6S                ; NO
5007 026154 104402 026160      TYPE    .+2                ; .ASCIZ " WC="
5008 026166 017746 152712      MOV     @RSWC,-(6)       ; PUT @RSWC ON STACK
5009 026172 104404                TYPE0   ; TYPE STACK IN OCTAL
5010 026174 032767 000040 176362 6S:    BIT     #DS,.HLTCT       ; DRIVE STATUS
5011 026202 001475                BEQ     PTDONE            ; NO
5012 026204 104402 026210      TYPE    .+2                ; .ASCIZ " DS="

```


5013	026216	017746	152670		MOV	ARSDS,-(6)		:PUT ARSDS ON STACK
5014	026222	104404			TYPE0			:TYPE STACK IN OCTAL
5015	026224	000167	000146		JMP	PTDONE		:GET OUT
5016	026230	042767	000200	176326	BIC	#200,.HLTCT	SEEC:	:CLEAR COMMON BIT
5017	026236	032767	000240	176320	BIT	#DT,.HLTCT		:TYPTXT DRIVE TYPE?
5018	026244	001410			BEQ	9\$:NO
5019	026246	104402	026252		TYPE	.+2		:.ASCIZ " DT="
5020	026260	017746	152642		MOV	ARSDT,-(6)		:PUT ARSDT ON STACK
5021	026264	104404			TYPE0			:TYPE STACK IN OCTAL
5022	026266	032767	000210	176270	BIT	#DB,.HLTCT	9\$:	:TYPTXT DATA BUFFER
5023	026274	001410			BEQ	10\$:NO
5024	026276	104402	026302		TYPE	.+2		:.ASCIZ " DB="
5025	026310	017746	152606		MOV	ARSDB,-(6)		:PUT ARSDB ON STACK
5026	026314	104404			TYPE0			:TYPE STACK IN OCTAL
5027	026316	032767	000220	176240	BIT	#MR,.HLTCT	10\$:	:TYPTXT MN?
5028	026324	001410			BEQ	11\$:NO
5029	026326	104402	026332		TYPE	.+2		:.ASCIZ " MR="
5030	026340	017746	152560		MOV	ARSMR,-(6)		:PUT ARSMR ON STACK
5031	026344	104404			TYPE0			:TYPE STACK IN OCTAL
5032	026346	032767	000204	176210	BIT	#LA,.HLTCT	11\$:	:TYPTXT LA?
5033	026354	001410			BEQ	PTDONE		:NO
5034	026356	104402	026362		TYPE	.+2		:.ASCIZ " LA="
5035	026370	017746	152524		MOV	ARSLA,-(6)		:PUT ARSLA ON STACK
5036	026374	104404			TYPE0			:TYPE STACK IN OCTAL
5037	026376	052767	100000	152562	BIS	#BIT15,ONCEE	PTDONE:	:SET FORND ERROR FLAG
5038	026404	032767	000040	152554	BIT	#BIT5,ONCEE		
5039	026412	001466			BEQ	1\$		
5040	026414	104402	026420		TYPE	.+2		:.ASCIZ <15><12>"MAINT CLOCK COUNT "
5041	026446	016767	152522	152552	MOV	MCCNT,WORK4		:GET MAINT CLOCK COUNT
5042	026454	016767	152516	152540	MOV	MCCNT+2,WORK2		:CAL NUMBERS FOR DOUBLE PRECISION
5043	026462	006167	152534		ROL	WORK2		
5044	026466	006167	152534		ROL	WORK4		
5045	026472	000241			CLC			
5046	026474	016746	152526		MOV	WORK4,-(6)		:PUT WORK4 ON STACK
5047	026500	104406			TYPES			:TYPE STACK IN OCTAL - SUPRESS
5048	026502	012767	000005	152520	MOV	#5,WORK5		
5049	026510	005067	152516		CLR	WORK6	2\$:	
5050	026514	006167	152502		ROL	WORK2		
5051	026520	006167	152506		ROL	WORK6		
5052	026524	006167	152472		ROL	WORK2		
5053	026530	006167	152476		ROL	WORK6		
5054	026534	006167	152462		ROL	WORK2		
5055	026540	006167	152466		ROL	WORK6		
5056	026544	016746	152462		MOV	WORK6,-(6)		:PUT WORK6 ON STACK
5057	026550	104406			TYPES			:TYPE STACK IN OCTAL - SUPRESS
5058	026552	005367	152452		DEC	WORK5		
5059	026556	001354			BNE	2\$		
5060	026560	104402	026564		TYPE	.+2		:.ASCIZ <15><12>
5061	026570	000207			RTS	PC	1\$:	
5062	026572	000300			.BLKW	300	INBUF:	
5063	027372	000300			.BLKW	300	OUTBUF:	
5064		000001			.END			

CS1 = 000001	798#	1025	1083	1123	1127	1131	1139	1356	1360	1438	1443		
CS2 = 000200	805#	1016	1079	1149	1159	1163	1168	1413					
DA = 000004	800#	1036	1251	1255	1259	1280	2180	2231	2384	2400	2641	2647	2851
	2857	3074	3080	3283	3480	3592	3599	3605	3692	3696	3702	3706	3942
	3951	5000											
DAO = 001000	826#												
DB = 000210	807#	5022											
DCK = 100000	827#												
DISPLA= 177570	725#	4681*	4685*										
DLT = 100000	820#												
DONE = 021134	989	4138#											
DONEE = 002316	971	977	989#										
DRY = 000200	821#												
DS = 000040	803#	1020	1838	1905	1909	1920	1924	2021	2025	2028	2128	2132	2135
	2175	2184	2188	2226	2235	2239	2280	2286	2290	2327	2333	2337	2393
	2851	3074	3283	3480	3486	3520	3538	3592	3599	3605	3611	3651	3655
	3661	3665	3687	3692	3702	3706	3934	3948	4037	4040	4045	4048	4090
	4105	4127	4131	5010									
DSCK = 104426	2003	2009	2016	2078	2084	2123	2153	2170	2204	2221	2258	2275	2307
	2322	4000	4903#										
	809#	5017											
DT = 000240	942	945#											
DVNUM = 001722	799#	1042	1100	1288	1292	1299	1305	1312	3538				
ER = 000002	824#												
ERR = 040000	757#	4703*	4723										
ERRORS = 001002	4367	4394#											
E1910 = 022620	4369	4396#											
E1915 = 022627	4363	4392#											
E192 = 022604	4365	4393#											
E197 = 022612	4357	4386#											
E2410 = 022557	4359	4388#											
E2412 = 022566	4361	4390#											
E2415 = 022575	4351	4380#											
E242 = 022535	4353	4382#											
E245 = 022543	4355	4384#											
E247 = 022551	4345	4374#											
E3010 = 022510	4347	4376#											
E3012 = 022517	4349	4378#											
E3015 = 022526	4339	4371#											
E302 = 022466	4341	4372#											
E305 = 022474	4343	4373#											
E307 = 022502	761#	4630	4631										
FILCHR = 001014	712#	715#	832#	897*	899	2424*	2648	2671*	2903*	3095*	3104	3139*	3302*
FLAG2 = 001144	3311	3337*	3720*	4071*	4409	4442	4452	4454*	4457*	4530	4543	4551	
	1189#												
FLOTBA = 003346	1264#												
FLOTDA = 003646	1227#												
FLOTWC = 003506	2587	2795	3017	3889	4529#								
GENCRC = 023454	2460	2708	2931	3143	3341	3757	4911#						
GETSP = 104446	751#	931*	1154*	1189#	1191	1193	1196*	1227*	1229	1231	1234*	1264*	1266
GOOD = %000001	1268	1271*	1367*	1368#	1370	1406*	1407*	1409	1414*	1416	1419*	1421	1607*
	1608	1619*	1620	1685#	1686	1696*	1697	1708*	1709	1720*	1721	1750*	1751
	1772*	1773	1783*	1784	1790*	1791	1829*	1830	1842*	1843	1851*	1852	1859*
	1860	1940*	1942	1954*	1955	1961*	1962	1968*	1969	2164*	2165	2215*	2216
	2269*	2270	2317*	2318	2360*	2361	2365*	2370	2371	2376*	2378	2402*	2864*
	2869	2875	2881	3528*	3529*	3530	3548*	3557*	3558*	3559	3643*	3644	4203*

MRINT = 104424	1464	1644	1822	1896	1936	1996	2045	2150	2201	2255	2304	2355	2431
MRLF1 006240	2668	2900	3102	3309	3511	3582	3633	3681	3729	3968	4902#		
MRLF2 006252	1815	1819#	1883										
MRLF3 006E52	1827#												
MROP 006736	1868#	1870	1872	1874	1876	1878	1880						
MROPER 007126	1891#												
MROPI 016450	1932#	1977											
MROPIA 016642	3626#												
MRPAR 016764	3654	3657#											
MRRD 012376	3676#												
MRRD1 012566	2662#												
MRRD2 012642	2724#	2733											
MRRT1 007716	2748#	2757											
MRSRC 007652	2054#	2063											
MRSRCH 007470	2040#												
MRTIME 004676	1991#												
MRTIM1 004744	1459#												
MRT2 005022	1485#	1494											
MRT2A 005032	1507	1512#											
MRT2B 005066	1514#	1523											
MRT2C 005122	1527#	1536											
MRT3 005240	1541#	1550											
MRT4 005432	1583#	1613											
MRT4A 005504	1636#												
MRT4B 005562	1654#	1669											
MRT4C 005614	1677#	1763											
MRT4D 005666	1694#	1701											
MRT4E 005674	1717#	1755											
MRT4F 006114	1718#	1725	1767										
MRT4G 006146	1730	1769#											
MRVR 020404	1781#												
MRVRR 020714	4068#												
MRVR1 020544	4104	4110#											
MRVR2 020720	4089	4092#											
MRWCK 013364	2650	4113#											
MRWCK1 013540	2894#												
MRWCK2 013614	2946#	2955											
MRWCK3 013716	2971#	2980											
MRWCK4 014116	2999#												
MRWCK5 014162	3048#												
MRWRT 011456	3071#												
MRWRT1 011624	2425#	4110											
MRWRT2 011700	2475#	2484											
MRWRT3 011764	2500#	2509											
MRWR1 010066	2526#	2535											
MULSYS 021214	2099#	2108											
MULTII 001672	4148	4154#	4805										
N = 000065	901	906	938#	4160									
	720#	997	1001#	1061	1065#	1111	1115#	1132	1133#	1141	1145#	1170	1171#
	1186	1187#	1198	1199#	1207	1208#	1223	1224#	1237	1238#	1244	1245#	1260
	1261#	1274	1275#	1281	1282#	1294	1295#	1300	1301#	1306	1307#	1313	1314#
	1323	1324#	1330	1331#	1338	1339#	1346	1350#	1362	1363#	1374	1375#	1386
	1387#	1398	1402#	1426	1430#	1445	1449#	1623	1627#	1797	1801#	1885	1889#
	1926	1930#	1979	1983#	2029	2033#	2136	2140#	2189	2193#	2241	2245#	2291
	2295#	2338	2342#	2411	2415#	2651	2655#	2884	2888#	3081	3085#	3292	3296#
	3495	3499#	3564	3568#	3614	3618#	3666	3670#	3708	3712#	3953	3957#	4061

N10

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
CROSS REFERENCE TABLE -- USER SYMBOLS

MACY11 27(732) 04-OCT-76 13:11 PAGE 132

RSAS	001116	776#	928	930	945	1058	1941	1960	2022*	2023	2129*	2130	2313*	2387
		3513*	4993											
RSBA	001106	772#	1004*	1028	1069*	1086	1173*	1174	1177*	1178	1181*	1182	1191*	1192
		1202*	1203*	1204	1388*	1390	1393*	1394	2457*	2700*	2928*	3134*	3332*	3515*
		3545	3547	3627*	3754*	3972*	4080*	4095*	4119*	4998				
RSBAB	001142	786#	1389*											
RSCS1	001100	769#	1003*	1023	1068*	1081	1120*	1121	1124*	1125	1128*	1129	1135*	1136*
		1137	1352*	1354	1357*	1358	1434*	1441	1827*	1900*	1916*	1949*	1952*	2002*
		2019	2026	2077*	2126	2133	2152*	2159*	2173	2186	2203*	2224	2237	2257*
		2278	2288	2306*	2325	2335	2369*	2382	2453*	2639	2702*	2849	2930*	3072
		3136*	3281	3334*	3478	3517*	3518	3536	3588*	3602	3635*	3653	3690	3700
		3756*	3940	3981*	4082*	4083	4088	4097*	4098	4103	4121*	4122	4129	4924
		4978												
RSCS1B	001134	783#	1353*	2390*										
RSCS2	001102	770#	924*	939*	943*	954*	1002*	1009*	1014	1017*	1067*	1074*	1076*	1077
		1080*	1147	1150*	1151	1153	1156*	1157	1160*	1161	1164*	1165*	1166	1169*
		1364*	1365*	1369	1408	2361*	2363	2371*	2375	2391	2395*	3527	3556	3682*
		4079*	4094*	4118*	4124	4914*	4915*	4986						
RSCS2B	001136	784#	1366*											
RSDA	001110	773#	1005*	1034	1070*	1092	1248*	1249	1252*	1253	1256*	1257	1266*	1267
		1276*	1277*	1278	1998*	2001*	2151*	2178	2202*	2210*	2229	2256*	2305*	2357*
		2372*	2396	2645	2855	3078	3587*	3683*	3694	3949	3970*	5003		
RSDB	001122	778#	1072*	1403*	1404*	1415	1420	5025						
RSDS	001112	774#	1018	1828	1850	1906	1921	1967	3596	3649	3662	4038	4046	4214
		5013												
RSDT	001126	780#	955	957	1569	1595	1733	1999	5020					
RSER	001114	775#	925*	940*	1006*	1040	1071*	1098	1285*	1286	1289*	1290	1296*	1297
		1302*	1303	1308*	1309*	1310	1841	1858	1902	1917	1939*	1953	2163	2182
		2214	2233	2264*	2268	2284	2316	2331	2362	3484	3512*	3589	3609	3642
		3658	3685	3704	3888*	3946	4035	4043	4982					
RSLA	001120	777#	1474	1506	1574	1606	1618	1658	1665	1684	1695	1707	1719	1732
		1771	1782	1789	5035									
RSMR	001124	779#	1007*	1046	1075*	1103	1317*	1318	1325*	1326*	1327	1332*	1333	1340*
		1341	2765*	2766*	2770*	2771*	2987*	2988*	2992*	2993*	3200*	3201*	3205*	3206*
		3397*	3398*	3402*	3403*	3990*	3991*	4196*	4197*	4202	4211*	4212*	4295	4318
		4436*	4438*	4467	4920*	4953	5030							
RSREG	025672	4713	4966#											
RSVCPS	001132	782#	1432*	3753*										
RSVEC	001130	781#	1431*	3752*										
RSWC	001104	771#	1008*	1052	1073*	1108	1211*	1212	1215*	1216	1219*	1220	1229*	1230
		1239*	1240*	1241	1376*	1378	1381*	1382	2458*	2642	2701*	2852	2929*	3075
		3135*	3284	3333*	3481	3516*	3540	3628*	3755*	3943	3971*	4077*	4093*	4117*
		5008												
RSMCB	001140	785#	1377*											
RSO	001156	837#	4566*	4574	4587									
RO	=%000000	727#	4144*	4782	4801*									
R1	=%000001	728#	938*	941*	4783	4793*	4794*	4800*	4819	4823*	4834*	4836*	4838*	4840
		4843*												
R2	=%000002	729#	915*	917*	922	923	1512*	1571*	1573*	1574	1600*	1603*	1605*	1607
		1640*	1685	1696	1706*	1708	1720	1729	1731*	1745*	1747*	1748*	1750	1759
		1772	1816*	1827	1833	1868*	1869	1871	1873	1875	1877	1879	1881	2440*
		2441*	2442*	2447*	2452*	2589*	2590*	2591*	2592	2679*	2680*	2681*	2686*	2690*
		2703*	2706*	2797*	2799*	2800*	2801	2911*	2912*	2913*	2918*	2922*	3019*	3021*
		3022*	3023	3121*	3123*	3126*	3127*	3129*	3137*	3141*	3229*	3230*	3231*	3232
		3320*	3322*	3325*	3326*	3328*	3335*	3339*	3426*	3427*	3428*	3429	3577*	3585*
		3587	3736*	3737*	3738*	3743*	3747*	3891*	3892*	3893*	3894	3988*	3990	4784

TST16 003676 1274#
TST17 003722 1281#
TST2 002566 1064#
TST20 003764 1294#
TST21 004006 1300#
TST22 004030 1306#
TST23 004054 1313#
TST24 004112 1323#
TST25 004140 1330#
TST26 004176 1338#
TST27 004234 1349#
TST3 003002 1114#
TST30 004306 1362#
TST31 004352 1374#
TST32 004422 1386#
TST33 004474 1401#
TST34 004602 1429#
TST35 004674 1448#
TST36 005430 1626#
TST37 006212 1800#
TST4 003074 1132#
TST40 006734 1888#
TST41 007124 1929#
TST42 007466 1982#
TST43 007650 2032#
TST44 010206 2139#
TST45 010372 2192#
TST46 010546 2244#
TST47 010720 2294#
TST5 003122 1144#
TST50 011134 2341#
TST51 011446 2414#
TST52 012374 2654#
TST53 013362 2887#
TST54 014220 3084#
TST55 015100 3295#
TST56 016030 3498#
TST57 016250 3567#
TST6 003262 1170#
TST60 016446 3617#
TST61 016762 3669#
TST62 017102 3711#
TST63 020076 3956#
TST64 020402 4064#
TST7 003344 1186#

TTRGG 002570 1066#
TYPE = 104402

TYPE0 = 104404

TYPES = 104406
UNCMP 001164

903	909	926	950	963	967	984	985	1832	1854	1863	1944	1957
1964	1971	2167	2168	2218	2219	2272	2273	2320	2408	2878	3487	3646
3660	4109	4126	4143	4155	4168	4172	4335	4339	4341	4343	4345	4347
4349	4351	4353	4355	4357	4359	4361	4363	4365	4367	4369	4702	4706
4712	4764	4804	4847	4867	4870	4874	4893#	4968	4971	4977	4981	4985
4992	4997	5002	5007	5012	5019	5024	5029	5034	5040	5060		
4711	4894#	4970	4973	4979	4983	4987	4994	4999	5004	5009	5014	5021
5026	5031	5036										
966	983	1835	2880	4170	4174	4895#	5047	5057				
840#	923*	946*	952	972*	978*	986*	1940	1961				

MAINDEC-11-DERSD-B
DERSOB.P11RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

ADC	4140	4199	4440												
ADD	1571	1573	1603	1605	1747	1868	2590	2799	3021	3230	3427	3892	4139	4198	4204
	4207	4216	4219	4247	4298	4321	4331	4404	4439	4470	4597	4641	4644	4889	4930
	4937	4956													
ASL	972														
BCC	1197	1235	1272												
BCE	973	2367	2446	2685	2917	3288	3491	3586	3742						
BEG	900	906	919	929	942	953	956	962	1015	1019	1024	1029	1035	1041	1047
	1053	1059	1078	1082	1087	1093	1099	1104	1109	1122	1126	1130	1138	1148	1152
	1158	1162	1167	1175	1179	1183	1194	1205	1213	1217	1221	1232	1242	1250	1254
	1258	1269	1279	1287	1291	1298	1304	1311	1321	1328	1336	1344	1355	1359	1371
	1379	1383	1391	1395	1410	1417	1422	1442	1475	1507	1570	1575	1596	1598	1609
	1621	1659	1666	1687	1698	1710	1722	1730	1734	1736	1752	1762	1774	1785	1792
	1831	1844	1853	1861	1870	1872	1874	1876	1878	1880	1882	1903	1907	1918	1922
	1943	1956	1963	1970	2000	2020	2024	2027	2127	2131	2134	2166	2174	2179	2183
	2187	2217	2225	2230	2234	2238	2271	2279	2285	2289	2319	2326	2332	2336	2379
	2383	2388	2392	2399	2643	2646	2850	2853	2856	2868	2870	2874	3073	3076	3079
	3282	3285	3479	3482	3485	3519	3531	3537	3541	3546	3560	3590	3597	3610	3645
	3650	3654	3659	3664	3686	3691	3695	3701	3705	3941	3944	3947	3950	4036	4039
	4044	4047	4125	4145	4229	4306	4311	4410	4443	4448	4450	4463	4480	4483	4531
	4544	4564	4572	4578	4585	4591	4596	4627	4658	4660	4675	4687	4701	4753	4756
	4826	4859	4928	4954	4991	4996	5001	5006	5011	5018	5023	5028	5033	5039	
BGT	4677	4828													
BHIS	4509														
BIC	715	897	898	951	968	1319	1334	1342	1599	1638	1744	1745	1749	1760	1813
	1840	1847	1857	1865	1892	1910	1925	1978	2146	2251	2427	2561	2586	2664	2896
	3098	3305	3576	3595	3601	3607	3677	3725	3858	3883	4076	4235	4309	4314	4437
	4446	4451	4457	4504	4512	4514	4519	4522	4562	4576	4589	4606	4607	4645	5016
BICB	4863														
BIS	712	907	994	1365	1368	1407	1460	1602	1637	1742	1766	1836	1845	1855	1862
	1914	1976	1992	2041	2053	2098	2376	2409	2426	2474	2562	2580	2585	2663	2723
	2793	2895	2945	3015	3097	3139	3158	3228	3304	3337	3355	3425	3529	3558	3591
	3598	3604	3682	3724	3771	3859	3877	3882	4079	4094	4118	4167	4282	4291	4293
	4307	4312	4316	4435	4444	4445	4454	4456	4461	4466	4481	4484	4501	4565	4573
	4579	4586	4592	4608	5037										
BISB	4757														
BIT	899	948	952	959	961	970	976	980	1046	1081	1098	1103	1327	1597	1735
	1740	1814	1898	1947	1974	2363	2391	2406	2573	2575	2648	2791	2871	3013	3104
	3226	3311	3423	3518	3583	3602	3870	3872	4068	4124	4141	4158	4165	4228	4237
	4329	4407	4409	4442	4447	4452	4462	4491	4493	4530	4543	4551	4571	4584	4623
	4657	4661	4672	4700	4704	4717	4744	4953	4988	4990	4995	5000	5005	5010	5017
	5022	5027	5032	5038											
BLOS	913	4496	4499	4554											
BLT	4830														
BMI	4084	4099	4925												
BNE	949	958	960	971	977	981	1412	1437	1494	1523	1536	1550	1586	1613	1617
	1669	1701	1725	1741	1755	1777	1815	1899	1948	1975	2013	2063	2076	2088	2108
	2364	2407	2454	2484	2509	2535	2574	2576	2578	2583	2596	2617	2649	2692	2707
	2733	2757	2790	2792	2836	2872	2924	2955	2980	3012	3014	3058	3105	3125	3131
	3142	3168	3192	3225	3227	3267	3312	3324	3330	3340	3365	3389	3422	3424	3464
	3526	3584	3603	3749	3781	3806	3832	3871	3873	3875	3880	3898	3916	3979	3993
	3995	4024	4069	4075	4086	4101	4116	4142	4159	4166	4206	4218	4238	4246	4276
	4279	4288	4290	4297	4320	4330	4403	4408	4453	4459	4465	4469	4492	4494	4503
	4511	4521	4552	4594	4624	4629	4635	4643	4662	4673	4705	4718	4745	4759	4761
	4795	4865	4872	4942	4952	4967	4989	5059							
BPL	2640	4089	4104	4123	4130	4633	4638	4715	4861						

H11

MAINDEC-11-DERSD-B
DERSDB.P11

RS11-RS04 MAINTENANCE MODE DIAGNOSTIC
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 04-OCT-76 13:11 PAGE 141

BR	716	921	944	969	975	988	1572	1601	1604	1743	1746	1950	2368	2404	2448
	2687	2804	2882	2919	3026	3235	3432	3612	3744	3935	4091	4128	4249	4266	4308
	4313	4333	4412	4432	4455	4475	4497	4513	4558	4599	4639	4663	4668	4671	4731
CLC	4733	4742	4808	4839	4848	4868	4929	4974							
CLR	1190	1228	1265	3489	4548	4939	5045								
	895	896	915	939	979	1136	1165	1203	1240	1277	1309	1326	1403	1414	1512
	1639	1640	1790	1859	2210	2313	2360	2372	2441	2443	2671	2680	2682	2705	2777
	2912	2914	2999	3123	3129	3140	3212	3322	3328	3338	3409	3737	3739	4008	4071
	4156	4157	4236	4239	4240	4250	4253	4260	4263	4274	4406	4413	4414	4417	4420
	4426	4429	4478	4535	4547	4555	4559	4570	4583	4734	4793	4823	4916	4917	4923
CLRB	4961	4962	5049												
CMP	1366	4719	4741	4743	4763	4873									
	912	928	955	957	1014	1018	1023	1052	1077	1108	1121	1125	1129	1137	1147
	1151	1157	1161	1166	1174	1178	1182	1193	1212	1216	1220	1231	1249	1253	1257
	1268	1286	1290	1297	1303	1320	1335	1343	1354	1358	1370	1378	1382	1390	1394
	1409	1416	1421	1440	1441	1569	1574	1595	1608	1620	1686	1697	1709	1721	1729
	1733	1751	1761	1773	1784	1791	1830	1843	1852	1860	1869	1871	1873	1875	1877
	1879	1881	1906	1921	1942	1955	1962	1969	1999	2019	2026	2126	2133	2165	2173
	2178	2186	2216	2224	2229	2237	2270	2278	2288	2318	2325	2335	2378	2382	2396
	2645	2849	2855	2869	2875	2881	3072	3078	3281	3478	3484	3530	3536	3540	3545
	3559	3589	3596	3644	3649	3653	3662	3685	3690	3694	3700	3940	3943	3946	3949
	4035	4038	4046	4205	4217	4296	4319	4449	4464	4468	4495	4498	4502	4508	4510
	4520	4553	4642	4669	4760	4858									
CMPB	905	4628	4659	4676	4827	4829	4864	4871							
DEC	920	941	1411	1436	1493	1522	1535	1549	1585	1612	1616	1668	1700	1724	1754
	1776	2012	2062	2075	2087	2107	2453	2483	2508	2534	2577	2582	2595	2616	2691
	2706	2732	2756	2789	2835	2867	2873	2923	2954	2979	3011	3057	3124	3130	3141
	3167	3191	3224	3266	3323	3329	3339	3364	3388	3421	3463	3523	3748	3780	3805
	3831	3874	3879	3897	3915	3978	3992	3994	4023	4074	4085	4100	4115	4256	4423
	4458	4593	4595	4634	4941	4951	5058								
EMT	721														
HALT	708	927	987	4716	4790	4807									
INC	943	974	1706	1731	2365	4490	4703	4794	4926						
INCB	4679	4684	4754	4758											
JMP	709	713	901	935	989	1439	1763	1767	1883	1977	2650	3290	3493	4070	4110
	4135	4148	4160	4161	4721	4805	4975	5015							
JSR	1904	1908	1919	1923	1958	1965	1972	2587	2795	3017	3289	3488	3492	3889	4146
	4294	4317	4334	4433	4473	4561	4575	4588	4713						
MOV	888	889	890	891	892	893	894	911	914	922	923	924	925	930	931
	938	940	945	946	947	954	965	978	982	986	995	996	1001	1002	1003
	1004	1005	1006	1007	1008	1009	1017	1066	1067	1068	1069	1070	1071	1072	1073
	1074	1075	1076	1080	1119	1120	1124	1128	1135	1150	1153	1154	1156	1160	1164
	1169	1173	1177	1181	1189	1191	1192	1202	1211	1215	1219	1227	1229	1230	1239
	1248	1252	1256	1264	1266	1267	1276	1285	1296	1302	1308	1317	1318	1325	1332
	1333	1340	1341	1352	1364	1367	1369	1376	1388	1404	1405	1406	1408	1415	1419
	1420	1431	1432	1433	1434	1435	1484	1513	1526	1540	1582	1583	1606	1607	1614
	1618	1619	1653	1684	1685	1693	1695	1696	1707	1708	1716	1717	1719	1720	1732
	1750	1759	1764	1765	1769	1771	1772	1782	1783	1789	1816	1827	1828	1829	1833
	1834	1841	1842	1850	1851	1858	1900	1901	1915	1916	1939	1941	1946	1949	1951
	1952	1953	1954	1960	1967	1968	1998	2001	2002	2007	2022	2052	2073	2077	2082
	2097	2129	2151	2152	2159	2163	2164	2202	2203	2214	2215	2256	2257	2264	2268
	2269	2305	2306	2316	2317	2357	2359	2361	2369	2370	2371	2375	2395	2401	2402
	2424	2440	2442	2447	2449	2451	2452	2457	2458	2459	2473	2499	2525	2554	2555
	2556	2561	2589	2591	2592	2594	2607	2679	2681	2686	2688	2689	2690	2700	2701
	2702	2703	2704	2722	2747	2765	2766	2770	2771	2778	2780	2781	2794	2797	2798
	2800	2801	2803	2826	2863	2864	2865	2866	2877	2879	2903	2911	2913	2918	2920

	2921	2922	2928	2929	2930	2944	2970	2987	2988	2992	2993	3000	3002	3003	3016
	3019	3020	3022	3023	3025	3048	3095	3106	3121	3122	3126	3127	3128	3134	3135
	3136	3137	3138	3157	3182	3200	3201	3205	3206	3213	3215	3216	3229	3231	3232
	3234	3256	3302	3313	3320	3321	3325	3326	3327	3332	3333	3334	3335	3336	3354
	3379	3397	3398	3402	3403	3410	3412	3413	3426	3428	3429	3431	3453	3512	3513
	3515	3516	3517	3522	3527	3528	3547	3548	3556	3557	3577	3578	3587	3588	3627
	3628	3635	3642	3643	3683	3720	3722	3723	3736	3738	3743	3745	3746	3747	3752
	3753	3754	3755	3756	3770	3796	3822	3851	3852	3853	3878	3888	3891	3893	3894
	3896	3906	3964	3970	3971	3972	3975	3976	3977	3981	3986	3987	3988	3989	3990
	3991	4014	4073	4077	4078	4080	4081	4082	4093	4095	4096	4097	4114	4117	4119
	4120	4121	4144	4169	4173	4196	4197	4202	4203	4211	4212	4214	4215	4230	4231
	4241	4243	4244	4248	4292	4295	4315	4318	4327	4328	4405	4411	4415	4434	4436
	4438	4441	4460	4467	4474	4529	4532	4533	4534	4542	4545	4546	4560	4566	4574
	4587	4598	4600	4605	4620	4621	4622	4625	4640	4646	4647	4648	4664	4665	4667
	4670	4680	4681	4685	4688	4707	4710	4730	4737	4738	4739	4740	4765	4766	4767
	4768	4780	4781	4782	4783	4784	4785	4786	4787	4788	4789	4792	4796	4797	4798
	4799	4800	4801	4802	4803	4817	4818	4819	4820	4821	4824	4840	4841	4842	4843
	4856	4857	4875	4886	4888	4890	4914	4915	4920	4936	4938	4946	4969	4972	4978
	4982	4986	4993	4998	5003	5008	5013	5020	5025	5030	5035	5041	5042	5046	5048
	5056														
MOV8	1289	1353	1357	1377	1381	1389	1393	1940	1961	2390	4630	4631	4636	4678	4709
	4732	4735	4762	4825	4862	4869									
NOP	3524	3525	3933	3952	4133										
RESET	1010	1146													
ROL	917	1196	1234	1271	2366	2445	2684	2916	3287	3490	3585	3741	4251	4252	4254
	4255	4261	4262	4264	4265	4332	4418	4419	4421	4422	4427	4428	4430	4431	4549
	4550	4556	4557	4746	4748	4750	4834	4835	4836	4837	4838	4940	5043	5044	5050
	5051	5052	5053	5054	5055										
ROLB	4747	4749	4751												
ROR	4831	4832	4833												
RTI	4194	4200	4208	4213	4220	4242	4257	4299	4322	4416	4424	4471	4649	4682	4689
	4720	4769	4844	4876	4918	4921	4955	4957	4963						
RTS	4171	4175	4336	4340	4342	4344	4346	4348	4350	4352	4354	4356	4358	4360	4362
	4364	4366	4368	4370	4485	4505	4515	4523	4601	4609	4931	4943	5061		
SEC	916	2444	2683	2915	3740										
SUB	1600	1748	2593	2779	2802	3001	3024	3214	3233	3411	3430	3895	4708	4887	
TRAP	4889	4892	4893	4894	4895	4896	4897	4898	4899	4900	4901	4902	4903	4904	4905
	4906	4907	4908	4909	4910	4911	4912								
TST	918	1028	1034	1040	1058	1086	1092	1204	1241	1278	1310	1474	1506	1658	1665
	1902	1917	2023	2130	2182	2233	2284	2331	2362	2387	2639	2642	2852	3075	3284
	3481	3609	3658	3704	4043	4088	4103	4129	4245	4275	4278	4286	4289	4305	4310
	4402	4479	4482	4563	4577	4590	4666	4686	4714	4927	4966				
TSTB	4083	4098	4122	4626	4632	4637	4674	4752	4755	4860	4924				
.ASCIZ	904	910	927	951	964	986	1833	1855	1864	1945	1958	1965	1972	2169	2220
	2274	2321	2409	2879	3488	3647	3661	4110	4127	4144	4156	4169	4177	4183	4336
	4371	4372	4373	4374	4376	4378	4380	4382	4384	4386	4388	4390	4392	4393	4394
	4396	4707	4713	4805	4848	4868	4969	4972	4978	4982	4986	4993	4998	5003	5008
	5013	5020	5025	5030	5035	5041	5061								
.BLKB	4878														
.BLKW	4771	5062	5063												
.ENABL	1	691													
.END	5064														
.ENDC	751	889	895	1000	1064	1114	1132	1144	1170	1186	1198	1207	1223	1237	1244
	1260	1274	1281	1294	1300	1306	1313	1323	1330	1338	1349	1362	1374	1386	1401
	1429	1448	1626	1800	1888	1929	1982	2032	2139	2192	2244	2294	2341	2414	2654
	2887	3084	3295	3498	3567	3617	3669	3711	3956	4064	4141	4143	4144	4149	4662

K11

MAINDEC-11-DERSD-8
DERSDB.P11

RS11-R504 MAINTENANCE MODE DIAGNOSTIC
CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MACY11 27(732) 04-OCT-76 13:11 PAGE 144

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*, DERSDB.SEQ/SOL/CRF/PAGNUM/NL: TQC/DS:ERFZ=SYSMAC.SML, DERSDB.P11
RUN-TIME: 29 48 7 SECONDS
RUN-TIME RATIO: 334/85=3.8
CORE USED: 22K (43 PAGES)

