

RH70-RS04

MAINT MODE DIAGNOSTIC
MD-11-DERSD-C

EP-DERSD-C-DL-B
COPYRIGHT 1977
FICHE 1 OF 1

JUN 1977
digital
MADE IN USA

The image displays a grid of 100 small diagnostic data tables, arranged in 10 rows and 10 columns. Each table contains various alphanumeric characters, numbers, and symbols, representing diagnostic data for the MD-11-DERSD-C system. The data is organized into columns and rows, with some tables showing headers and sub-headers. The overall layout is a dense grid of diagnostic information.

A small diagnostic data table located in the bottom right corner of the page, containing a few lines of alphanumeric characters and numbers.

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90

TABLE OF CONTENTS

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

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

DESCRIPTION

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

2. REQUIREMENTS

2.1 EQUIPMENT

146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
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

DESCRIPTION

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

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 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

201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
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

DESCRIPTION

TYPE UNIT #

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

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

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

2. STARTING ADDRESS 220

- A. SET SWITCHES (SEE SECTION 5)
- 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
- SW<09> = 0 BELL ON PASS COMPLETE
- SW<09> = 1 LOOP ON ERROR
- SW<08> = 1 LOOP ON TEST IN SW<7:0>

5.1 SUBROUTINE ABSTRACTS

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

5.1.1 CLRDK

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

257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311

DESCRIPTION

THE NEXT INSTRUCTION FOLLOWING THE CLRDK INSTRUCTION.

5.1.2 MRDMD

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

5.1.3 MRINT

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

5.1.4 MRIND

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

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.

312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367

DESCRIPTION

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 MRDB (BIT 12) AND MRDT (BIT 14) IN BITS IN RSMR WHEN WRITING. THIS ROUTINE MAKES BITS 16 AND 17 OF EACH DATA WORD (RS04 WRITES 18 BIT WORDS) EQUAL ZERO. THE PROGRAM RETURNS TO THE NEXT INSTRUCTION FOLLOWING THE XBIT INSTRUCTION.

5.1.9 CLKD1 AND 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 MRDB (BIT 12) AND MRDT (BIT 14) IN RSMR WHEN WRITING. THESE ROUTINES ALSO CALCULATE THE CORRECT STATES OF THE CRCH, 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 CRCH 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.

368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420

DESCRIPTION

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

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

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

421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475

DESCRIPTION

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

4. TEST "CLEAR BIT" IN RSCS2

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

5. LOAD RSOB WITH ALL ONES AND ALL ZEROS

476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
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

DESCRIPTION

LOAD BSDB 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,
ETC.

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

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

DESCRIPTION

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

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

531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
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
607
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

DESCRIPTION

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

16. MAINTENANCE READ TEST

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

17. MAINTENANCE MODE DATA WRITE CHECK TEST

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

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

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

643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
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

DESCRIPTION

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

IN THIS TEST THE INTERRUPT ENABLE (I.E.,) BIT IS SET. A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMA" 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-DERSD-C RH70-RSD4 MAINTENANCE MODE DIAGNOSTIC
 ; COPYRIGHT 1974, 1975, 1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
 ; PROGRAM BY STANLEY KARACKIEWICZ

! SWITCH USE
 ; -----

```

699          100000          SW15= 100000          ;HALT ON ERROR
700          040000          SW14= 40000           ;LOOP ON TEST
701          020000          SW13= 20000           ;INHIBIT ERROR TYPEOUTS
702          010000          SW12= 10000           ;TYPEOUT ALL ERRORS IN DATA COMPARE ROUTINE
703          004000          SW11= 4000            ;RUN MAINTENANCE MODE VERIFY TEST
704          002000          SW10= 2000            ;0 - BELL ON PASS COMPLETE
705                                     ;1 - BELL ON ERROR
706          001000          SW9= 1000             ;LOOP ON ERROR
707          000400          SW8= 400              ;LOOP ON TEST IN SW(7:0)
708          000000          . = 0                  ;TRAP CATCHER FROM 0 - 776
709          000046          . = 46                 ;HOOKS FOR ACT 11
710 000046 021200          SENDAD
711          000052          . = 52
712 000052 040000          BIT14
713          000200          . = 200
714 000200 000137 000232          JMP          2#BEGIN1
715
716          000220          . = 220
717 000220 052767 000100 000716          . = 220          #BIT6 FLAG2          ;TEST ALL DRIVES
718 000226 000137 001234          BEGIN2: JMP          2#BEGIN
719
720 000232 042767 000100 000704          BEGIN1: BIC          #BIT6 FLAG2          ;CLEAR MULTI DRIVE FLAG
721 000240 000772          BR          BEGIN2
722
723

```

```

724
725 000001
726 104000
727 177776
728 177776
729 177570
730 177570
731 000007
732 000000
733 000001
734 000002
735 000003
736 000004
737 000005
738 000006
739 000007
740 000001
741 000002
742 000004
743 000010
744 000020
745 000040
746 000100
747 000200
748 000400
749 001000
750 002000
751 004000
752 010000
753 020000
754 040000
755 100000
756 000001
757 000000
758

```

```

N= 1
HLT= EMT
PS= 177776
PSW= PS
SMR= 177570
DISPLAY=SMR
BELL= 7
R0= X0
R1= X1
R2= X2
R3= X3
R4= X4
R5= X5
SP= X6
PC= X7
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= X1
BAD= X0

```

```

;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

```



```

759      001000
760
761 001000 000000
762 001002 000000
763 001004 000000 000000
764 001010 000000
765 001012 000000
766 001014 001000
767 001016 177564
768 001020 177566
769
770      001100
771
772
773
774 001100 172040
775 001102 172050
776 001104 172042
777 001106 172044
778 001110 172046
779 001112 172052
780 001114 172054
781 001116 172056
782 001120 172060
783 001122 172062
784 001124 172064
785 001126 172066
786 001130 000204
787 001132 000206
788 001134 172041
789 001136 172051
790 001140 172043
791 001142 172045
792
  
```

```

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

```

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

```

.=      1100
;DISK I/O REGISTERS
  
```

```

RSCS1: 172040
RSCS2: 172050
RSMC:  172042
RSBA:  172044
RSDA:  172046
RSDS:  172052
RSER:  172054
RSAS:  172056
RSLA:  172060
RSD8:  172062
RSMR:  172064
RSOT:  172066
RSVEC: 204
RSVCPS: 206
RSCS1B: 172041
RSCS2B: 172051
RSMCB:  172043
RSBAB:  172045
  
```

```

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

793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834

```
;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
```

000001	CS1=1	:CONTROL AND STATUS 1
000002	ER=2	:CONTROL AND STATUS 2
000004	DA=4	:DESIRED ADD
000010	WC=10	:WORD COUNT
000020	BA=20	:BUS ADDRESS
000040	DS=40	:DRIVE STATUS
000100	AS=100	:ATTENTION SUMMARY
000200	CS2=200	:CONTROL AND STATUS REG
000204	LA=204	:LOOK AHEAD
000210	DB=210	:DATA BUFFER
000220	MR=220	:MAINTENANCE
000240	DT=240	:DRIVE TYPE

;BIT ASSIGNMENTS FOR THE REGISTER BITS

040000	TRE=40000	:TRANSFER ERROR CS1
100000	SC=100000	:SPECIAL CONDITIONS CS1
000100	IR=100	:INPUT READY CS2
000200	OR=200	:OUTPUT READY CS2
002000	PGE=2000	:PROGRAM ERROR-CS2
010000	NED=10000	:NON-EXISTENT DRIVE CS2
040000	WCE=40000	:WRITE CHECK ERROR-CS2
100000	DLT=100000	:DATA LATE ERROR CS2
000200	DRY=200	:DRIVE READY DS
020000	PIP=20000	:POSITIONING IN PROGRESS DS
002000	LBT=2000	:LAST BLOCK TRANSFER-DS
040000	ERR=40000	:ERROR DS
100000	ATA=100000	:ATTENTION ACTIVE-DS
001000	DAO=1000	:DISK OVERFLOW ERROR-ER
100000	DCK=100000	:DATA CHECK ERROR-ER
000010	BAI=10	:BUS ADDR INCREMENT INHIBIT
000100	IE=100	:INTERRUPT INABLE CS1

```

835
836
837 001144 000000
838 001146 000000
839 001150 000000
840 001152 000000
841 001154 000000
842 001156 000000
843 001160 000000
844 001162 000000
845 001164 000000
846 001166 000000
847 001170 000000
848
849 001172 000000 000000
850 001174 000000
851 001200 000000
852 001202 000000
853 001204 000000
854 001206 000000
855 001210 000000
856 001212 000000
857 001214 000000
858 001216 000000
859 001220 000000
860 001222 000000
861 001224 000000
862 001226 000000
863 001230 000000
864 001232 000000

```

;WORKING LOCATIONS

```

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

```

```

;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
;CLOCK COUNTER FOR EACH WORD
;USED IN CRC CAL ROUTINE
;USED IN CRC CAL ROUTINE

```

```
865 ;DISCRIPTION OF BITS IN LOCATION ONCEE
866
867 ;BIT0 MEANS FOUND DRIVE
868 ;BIT1 ERROR DO NOT CHANGE ILLEGAL FUNCTION
869 ;BIT2 ERROR FLAG
870 ;BIT3 TESTING CODE 21 FLAG
871 ;BIT4 TEST ONLY ONE DRIVE
872 ;BIT5 TYPEOUT CLOCK COUNT
873 ;BIT6 1ST TRANSFER WORD FLAG
874 ;BIT7 WRITTING LAST WORD OF SECOTR
875 ;BIT8 TRANSFERRING CRC WORD
876 ;BIT9 FOR INTERLEAVED DRIVES
877 ;BIT10 1ST TIME FLAG IN SECTOR FRACTION TEST
878 ;BIT11 DO TKSEL TEST
879 ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
880 ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
881 ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
882 ;BIT15 MEANS ERROR FOUND
883
884 ;DISCRIPTION OF BITS IN LOCATION FLAG2
885
886 ;BIT0 SWITCH FOR RWCLK IN MR REG
887 ;BIT1 MAINTENANCE MODE VERIFY TEST
888 ;BIT2 IN WRITE CK TEST FOR CLKRI ROUTINE
889 ;BIT3 DONE 1ST CRC WD IN CRC TEST
890 ;BIT4 1ST TIME THROUGH IN CRC TEST
891 ;BIT5 IN CRC TEST
892 ;BIT6 FLAG TO TEST ALL DRIVES
```

H02

```

893 001234 012706 000500      BEGIN:  MOV     #500,SP           ;SET STACK TO *** 500 ***
894 001240 012737 025014 000024  MOV     @POWER,@#24      ;SET UP PF VECTOR
895 001246 012737 000340 000026  MOV     @340,@#26        ;LOCK OUT THE WORLD
896 001254 012737 024432 000030  MOV     @HLT,@#30        ;SET EMT VECTOR
897 001262 012737 000340 000032  MOV     @340,@#32        ;LOCK UP
898 001270 012737 025416 000034  MOV     @TRAP,@#34       ;SET TRAP VECTOR
899 001276 012737 000340 000036  MOV     @340,@#36        ;LOCK UP
900 001304 005067 177470           CLR     ICNT             ;INIT ICNT
901 001310 005067 177474           CLR     LAD              ;INIT LAD
902 001314 042767 177677 177622  BIC     @177677,FLAG2    ;
903 001322 042767 153777 177636  BIC     @153777,ONCEE    ;
904 001330 032767 000100 177606  BIT     @BIT6,FLAG2      ;TEST ALL DRIVES?
905 001336 001402           BEQ     5$              ;ASK
906 001340 000137 001672           JMP     @MULTII
907 001344           5$:
908 001344 104402 001350           TYPE   ,.+2             ;.ASCIZ <15><12>"TEST ALL DRIVES? (Y OR N) "
909 001406 104412           RDLIN  ,.+2             ;
910 001410 122767 000131 023760  CMPB   @Y,INPUT         ;TEST FOR YES
911 001416 001525           BEQ     MULTII          ;YES
912 001420 052767 000020 177540  BIS    @BIT4,ONCEE      ;SET TEST ONLY ONE DRIVE FLAG
913 001426           1$:
914 001426 104402 001432           TYPE   ,.+2             ;.ASCIZ "TYPE UNIT #"
915 001446 104410           RDOCT  ,.+2             ;
916 001450 012604           MOV     (6)+,R4         ;GET NUMBER
917 001452 022704 000010  CMP     @10,R4          ;CORRECT #
918 001456 101763           BLOS   1$              ;NO
919 001460 010467 177474           MOV     R4,UNNUM       ;SET UNIT #
920 001464 005002           CLR     R2              ;CLEAR WORK AREA
921 001466 000261           SEC     R2              ;SET CARRY
922 001470 006102           ROL     R2              ;SET WORK BIT
923 001472 005704           TST    R4              ;IS THIS BIT CORRESPOND WITH CORRECT DRIVE #
924 001474 001402           BEQ     3$              ;YES
925 001476 005304           DEC     R4              ;NO TRY AGAIN
926 001500 000773           BR     2$              ;TEST AGAIN
927 001502 010267 177454           MOV     R2,UNITSV      ;SET DRIVE BIT IN UNITSV
928 001506 010267 177452           MOV     R2,UNCOMP      ;SET UNIT COMPARE
929 001512 016777 177442 177362  MOV     UNUM,@RSAS2     ;LOAD DRIVE
930 001520 012777 177777 177366  MOV     #-1,@RSER      ;LOAD ERRORS
931 001526 104402 001532           TYPE   ,.+2             ;.ASCIZ "ALL ERROR LIGHTS ON SELECTED UNIT SHOULD BE ON
932 001642 000000           HALT                    ;WAIT FOR LIGHTS TO BE CHECKED
933 001644 026777 177312 177244  CMP     UNITSV,@RSAS   ;DID CORRECT ATA SET
934 001652 001405           BEQ     4$              ;
935 001654 017700 177236           MOV     @RSAS,BAD      ;GET RSAS
936 001660 016701 177276           MOV     UNITSV,GOOD    ;GET CORRECT AND
937 001664 104000           HLT                    ;RSAS=BAD GOOD=CORRECTIONS
938                                     ;ATA BIT SHOULD SET FOR ERRORS
939                                     ;WERE SET IN RSER
940 001666 000167 000430           4$:  JMP     NOWGO          ;START TESTING
  
```

```

941                                     ;NOW TEST FOR DRIVES
942
943 001672 012701 000010 MULTII: MOV #8,R1 ;PUT 8 INTO R1 FOR COUNT
944 001676 005077 177200 CLR #RSCS2 ;SET DEVICE TO ZERO
945 001702 012777 177777 177204 TRY: MOV #-1,RSER ;CAUSE AN ERROR +SETS BIT IN RSAS REG
946 001710 005301 DEC R1 ;DO A MAXIMUM OF 8 TIMES
947 001712 001403 BEQ DVNUM ;TESTED FOR ALL DRIVES GET OUT
948 001714 005277 177162 INC #RSCS2 ;INCREMENT DRIVE UNIT
949 001720 000770 BR TRY ;REPEAT FOR NEXT DRIVE
950 001722 017767 177170 177232 DVNUM: MOV #RSAS,UNITSV ;SAVE
951 001730 012767 000401 177226 MOV #401,UNCMP ;SETUP TO CMP WITH UNITSV
952 001736 012767 000000 177214 MOV #0,UNNUM ;PUT 0 INTO UNIT NO.
953 001744 032767 020000 175616 BIT #BIT13,SWR ;INHIBIT TYPE OUT?
954 001752 001015 BNE STTEST ;YES
955 001754 104402 001760 TYPE ,+2 ;.ASCIZ <15><12>"TESTING UNIT "
956 002000 042767 100000 177160 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
957 002006 036767 177152 177146 STTEST: BIT UNCMP,UNITSV ;IS THIS DRIVE ON THE SYSTEM
958 002014 001440 BEQ TRYNX ;NO
959 002016 016777 177136 177056 MOV UNNUM,#RSCS2 ;YES PUT UNIT # INTO CS2
960 002024 022777 000002 177074 3$: CMP #2,#RSDT ;IS THIS A RS04?
961 002032 001404 BEQ 1$ ;YES
962 002034 022777 000003 177064 CMP #3,#RSDT ;IS IT A RS04?
963 002042 001025 BNE TRYNX ;GET A NEW NUMBER
964 002044 032767 020000 175516 1$: BIT #BIT13,SWR ;INHIBIT TYPE OUT?
965 002052 001020 BNE 4$ ;YES
966 002054 032767 100000 177104 BIT #BIT15,ONCEE ;ANY ERRORS?
967 002062 001404 BEQ 5$ ;NO
968 002064 104402 002070 TYPE ,+2 ;.ASCIZ <15><12><12>
969 002074 5$:
970 002074 016746 177060 MOV UNNUM,-(6) ;PUT UNNUM ON STACK
971 002100 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
972 002102 104402 000040 TYPE ,40 ;TYPE SPACE
973 002106 042767 100000 177052 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
974 002114 000502 4$: BR NOWGO ;NOW TEST
975 002116 032767 000020 177042 TRYNX: BIT #BIT4,ONCEE ;MULTI DRIVE
976 002124 001074 BNE DONEE ;NO
977 002126 006367 177032 1$: ASL UNCMP ;CHECK NEXT BIT FOR DRIVE
978 002132 103403 BCS CHCKDV ;DID WE TEST ANY REG?
979 002134 005267 177020 INC UNNUM ;INC UNIT #
980 002140 000722 BR STTEST ;CHECK FOR NEXT DRIVE
    
```

```

981 002142 032767 000001 177016 CHCKDV: BIT      #BIT0,ONCEE    ;DID WE TEST ANY DRIVES?
982 002150 001062          BNE      DONEE      ;YES WE DID TEST A DRIVE
983 002152 012767 100000 177004      MOV      #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
984 002160 005067 176774          CLB      UNNUM      ;ANY AS BITS, THIS DEFAULTS TO
985 002164 032767 020000 175376      BIT      #BIT13,SMR  ;INHIBIT TYPE OUT?
986 002172 001050          BNE      4$        ;YES
987 002174 016746 176760          MOV      UNNUM,-(6) ;PUT UNNUM ON STACK
988 002200 104406          TYPES          ;TYPE STACK IN OCTAL - SUPRESS
989 002202 104402 000040          TYPE      ,40      ;TYPE SPACE
990 002206 104402 002212          TYPE      ,+2      ;.ASCIZ <15><12>"COULD NOT FIND DRIVE WILL TEST DRIVE 0
991 002304 012767 000001 176652      MOV      #1,UNCMP
992 002312 000000          HALT
993 002314 000402          4$: BR      NOWGO   ;WAIT
994 002316 000167 016612      DONEE: JMP     DONE   ;TEST DRIVE 0
995                                     ;GET OUT
996                                     ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
997                                     ;TO CLEAR ALL THE RH AND RS REGISTERS
998
999 002322 052767 000001 176636 NOWGO: BIS      #BIT0,ONCEE ;SET FOUND DRIVE FLAG
1000 002330 016767 022074 176632      MOV      TIMES,TIMSV ;SAVE TIME
1001 002336 012767 000001 022064      MOV      #1,TIMES   ;ONLY TEST ONCE
1002                                     ;*****
1003                                     ;TEST 1 RESET TEST FOR REGISTERS
1004                                     ;*****
1005 002344 104400          TST1: SCOPE
1006 002346 012737 000340 177776      MOV      #340,#PS   ;LOCK OUT INTERUPTS
1007 002354 016777 176600 176520      MOV      UNNUM,RS2   ;LOAD UNIT #
1008 002362 012777 177776 176510      MOV      #177776,RS1 ;SET ALL
1009 002370 012777 177777 176510      MOV      #177777,RSBA ;POSSIBLE R/W
1010 002376 012777 177777 176504      MOV      #177777,RSDA ;BITS IN THESE REGISTERS
1011 002404 012777 177777 176502      MOV      #177777,RSER
1012 002412 012777 177777 176504      MOV      #177777,RSMR
1013 002420 012777 177777 176456      MOV      #177777,RSWC
1014 002426 012777 177737 176446      MOV      #177737,RS2
1015 002434 000005          RESET          ;CLEAR ALL BITS IN ALL REG.
1016
1017                                     ;TEST RSCS2 FOR CLEARED BITS
1018
1019 002436 022777 000100 176436      CMP      #100,RS2    ;DID THESE BITS GET CLEARED?
1020 002444 001401          BEQ      +4        ;YES
1021 002446 104200          HLT      !CS2      ;(417) SHOULD BE CLEARED IN CS2
1022 002450 016777 176504 176424      MOV      UNNUM,RS2   ;PUT # OF UNIT IN TEST IN CS2
1023 002456 022777 010600 176426      CMP      #10600,RS2  ;IS DPR AND MOL SET?
1024 002464 001401          BEQ      +4        ;YES
1025 002466 104040          HLT      !DS       ;NO WHY NOT?
1026
1027                                     ;TEST CONTROL AND STATUS REG 1
1028 002470 022777 004200 176402      CMP      #4200,RS1   ;DID THE READY BIT SET?
1029 002476 001401          BEQ      +4        ;YES
1030 002500 104001          HLT      !CS1      ;READY SHOULD BE SET

```

K02

```

1031 ;TEST BUS ADDRESS REGISTER
1032
1033 002502 005777 176400 TST @RSBA ; IS BA REG. CLEARED
1034 002506 001401 BEQ +4 ; YES
1035 002510 104020 HLT !BA ; SHOULD BE 0
1036
1037 ;TEST DISK ADDRESS REGISTER
1038
1039 002512 005777 176372 TST @RSDA ; IS DA CLEARED
1040 002516 001401 BEQ +4 ; YES
1041 002520 104004 HLT !DA ; SHOULD BE 0
1042
1043 ;TEST ERROR REG RSER
1044
1045 002522 005777 176366 TST @RSER ; DID RSER CLEAR?
1046 002526 001401 BEQ +4 ; YES
1047 002530 104002 HLT !ER ; BITS(157015) SHOULD BE CLEARED
1048
1049 ;TEST RS MAINTENANCE REGISTER
1050
1051 002532 032777 000077 176364 BIT #77,@RSMR ; DID THESE BITS GET CLEARED
1052 002540 001401 BEQ +4 ; YES
1053 002542 104220 HLT !MR ; BITS(77) SHOULD BE 0
1054
1055 ;TEST WC REG IT SHOULD NOT CHANGE
1056
1057 002544 022777 177777 176332 CMP #177777,@RSWC ; DID IT CHANGE?
1058 002552 001401 BEQ +4 ; NO
1059 002554 104010 HLT !WC ; RESET SHOULD NOT MODIFY RSWC
1060
1061 ;TEST RSAS
1062
1063 002556 005777 176334 TST @RSAS ; IS REG CLEAR
1064 002562 001401 BEQ +4 ; YES
1065 002564 104100 HLT !AS ; NO
  
```


L02

MAINDEC-11-DERSO-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 25
 DERSDC.SRC 17-MAY-77 14:16 TST2 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS

```

1066 :*****
1067 :TEST 2            TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
1068 :*****
1069 002566    104400    TST2:    SCOPE
1070
1071 002570    012737    000340    177776    TTAGG:    MOV        #340, @RPS            ;LOCK OUT INTERRUPTS
1072 002576    016777    176356    176276            MOV        UNNUM, @RSCS2
1073 002604    012777    043576    176266            MOV        #43576, @RSCS1        ;SET ALL
1074 002612    012777    177777    176266            MOV        #177777, @RSBA       ;POSSIBLE
1075 002620    012777    177777    176262            MOV        #177777, @RSDA       ;REGISTERS
1076 002626    012777    177017    176260            MOV        #177017, @RSER
1077 002634    012777    177777    176260            MOV        #177777, @RSD8
1078 002642    012777    177777    176234            MOV        #177777, @RSWC
1079 002650    012777    020417    176224            MOV        #20417, @RSCS2
1080 002656    012777    000071    176240            MOV        #71, @RSMR
1081 002664    012777    000040    176210            MOV        #40, @RSCS2        ;CLEAR ALL BITS
1082 002672    022777    000100    176202            CMP        #100, @RSCS2       ;DID THE RIGHT BITS CLEAR?
1083 002700    001401                                                            BEQ        +4                ;YES
1084 002702    104200                                                            HLT        !CS2               ; (417) SHOULD BE CLEARED IN CS2
1085 002704    016777    176250    176170            MOV        UNNUM, @RSCS2       ;GET DRIVE NUMBER
1086 002712    032777    173577    176160            BIT        #173577, @RSCS1     ;DID ALL BITS GET CLEARED
1087 002720    001401                                                            BEQ        +4                ;YES
1088 002722    104001                                                            HLT        !CS1               ;NO, ALL BITS SHOULD BE 0
1089                                                            ;TEST BUS ADDRESS REGISTER
1090
1091                                                            TST        @RSBA             ;IS BA REG. CLEARED
1092                                                            BEQ        +4                ;YES
1093                                                            HLT        !BA               ;SHOULD BE 0
1094
1095                                                            ;TEST DISK ADDRESS REGISTER
1096
1097                                                            TST        @RSDA             ;IS DA CLEARED
1098                                                            BEQ        +4                ;YES
1099                                                            HLT        !BA               ;SHOULD BE 0
1100
1101                                                            ;TEST ERROR REG RSER
1102
1103                                                            BIT        #177777, @RSER     ;DID THESE BITS GET CLEARED
1104                                                            BEQ        +4                ;YES
1105                                                            HLT        !ER               ;BITS(157015) SHOULD BE CLEARED
1106
1107                                                            ;TEST RS MAINTENANCE REGISTER
1108                                                            BIT        #77, @RSMR          ;DID THESE BITS GET CLEARED
1109                                                            BEQ        +4                ;YES
1110                                                            HLT        !MR               ;BITS(77) SHOULD BE 0
1111
1112                                                            ;TEST WC REG. IT SHOULD NOT CHANGE
1113                                                            CMP        #177777, @RSWC     ;DID WC CHANGE
1114                                                            BEQ        +4                ;NO
1115                                                            HLT        !WC               ;WHY DID IT CHANGE?

```


N02

```

1161 003170 012777 020025 175704 1S:  MOV      #20025,RS2
1162 003176 022777 000125 175676      CMP      #125,RS2
1163 003204 001401      BEQ      +4
1164 003206 104200      HLT      CS2
1165 003210 012777 000012 175664      MOV      #12,RS2
1166 003216 022777 000112 175656      CMP      #112,RS2
1167 003224 001401      BEQ      +4
1168 003226 104200      HLT      CS2
1169 003230 012777 177777 175644      MOV      #-1,RS2
1170 003236 005077 175640      CLR      RS2
1171 003242 022777 000100 175632      CMP      #100,RS2
1172 003250 001401      BEQ      +4
1173 003252 104200      HLT      CS2
1174 003254 016777 175700 175620      MOV      UNNUM,RS2
1175 003262 104400
1176
1177
1178 003264 012777 177777 175614      MOV      #177777,RSBA
1179 003272 022777 177776 175606      CMP      #177776,RSBA
1180 003300 001401      BEQ      +4
1181 003302 104020      HLT      BA
1182 003304 012777 125252 175574      MOV      #125252,RSBA
1183 003312 022777 125252 175566      CMP      #125252,RSBA
1184 003320 001401      BEQ      +4
1185 003322 104020      HLT      BA
1186 003324 012777 052524 175554      MOV      #52524,RSBA
1187 003332 022777 052524 175546      CMP      #52524,RSBA
1188 003340 001401      BEQ      +4
1189 003342 104020      HLT      BA
1190
1191 003344 104400
1192
1193
1194 003346 012701 000002      FLOTBA: MOV      #2,GOOD
1195 003352 000241      CLC
1196 003354 010177 175526      1S:  MOV      GOOD,RSBA
1197 003360 017700 175522      MOV      RSBA,BAD
1198 003364 020100      CMP      GOOD,BAD
1199 003366 001401      BEQ      +4
1200 003370 104000      HLT
1201 003372 006101      ROL      GOOD
1202 003374 103367      BCC      1S

```

```

:SET THESE BITS
:DID THESE BITS GET SET
:YES
:NO CS2 SHOULD = 20125
:LOAD THESE BITS
:DID THESE BITS GET SET IN CS2
:YES
:BAD = CS2 GOOD = CORRECT ANS
:SET BITS
:CLEAR THEM
:DID CLEAR WORK
:YES
:R/W BITS DID NOT CLEAR
:GET UNIT #

:SET THE BITS
:DID THEY SET
:YES
:BITS 17776 SHOULD BE SET
:SET THESE BITS
:ARE THEY =
:YES
:SHOULD BE 125252
:SET THESE BITS
:ARE THEY =
:YES
:SHOULD BE 52524

:GET A 2
:CLEAR CARRY
:FLOAT NUMBER
:GET BA
:COMPARE BA
:BA CORRECT
:BAD=BA GOOD=CORRECT ANS
:ROTATE NUMBER
:LOOP TILL DONE

```

TST6: SCOPE
 ;CAN WE SET ALL THE RSBA BITS

TST7: SCOPE
 ;FLOAT A 1 THROUGH RSBA

FLOTBA:

1S:

B03

```

1203 003376 104400      TST10: SCOPE
1204
1205      ;CLEAR THE RSBA REGISTER
1206
1207 003400 012777 177777 175500      MOV      #177777, @RSBA      ;SET RSBA EQUAL TO ALL ONES
1208 003406 005077 175474      CLR      @RSBA
1209 003412 005777 175470      TST      @RSBA      ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1210 003416 001401      BEQ      +4      ;YES
1211 003420 104020      HLT      !BA      ;NO
1212 003422 104400      TST11: SCOPE
1213
1214      ;CAN WE SET ALL BITS IN RSMC REGISTER
1215
1216 003424 012777 177777 175452      MOV      #177777, @RSMC      ;SET MC BITS
1217 003432 022777 177777 175444      CMP      #177777, @RSMC      ;ARE ALL BITS SET
1218 003440 001401      BEQ      +4      ;YES
1219 003442 104010      HLT      !MC      ;NO
1220 003444 012777 125252 175432      MOV      #125252, @RSMC      ;SET THESE BITS
1221 003452 022777 125252 175424      CMP      #125252, @RSMC      ;ARE THEY =
1222 003460 001401      BEQ      +4      ;YES
1223 003462 104010      HLT      !MC      ;SHOULD BE 125252
1224 003464 012777 052525 175412      MOV      #52525, @RSMC      ;SET THESE BITS
1225 003472 022777 052525 175404      CMP      #52525, @RSMC      ;ARE THEY =
1226 003500 001401      BEQ      +4      ;YES
1227 003502 104010      HLT      !MC      ;SHOULD BE 152525
1228 003504 104400      TST12: SCOPE
1229
1230      ;FLOAT A 1 THROUGH RSMC
1231
1232 003506 012701 000001      FLOTMC: MOV      #1, GOOD      ;GET A 1
1233 003512 000241      CLC      ;CLEAR CARRY
1234 003514 010177 175364      IS:      MOV      GOOD, @RSMC      ;FLOAT NUMBER
1235 003520 017700 175360      MOV      @RSMC, BAD      ;GET MC
1236 003524 020100      CMP      GOOD, BAD      ;COMPARE MC
1237 003526 001401      BEQ      +4      ;MC CORRECT
1238 003530 104000      HLT      ;BAD=MC GOOD=CORRECT ANS
1239 003532 006101      ROL      GOOD      ;ROTATE NUMBER
1240 003534 103367      BCC      IS      ;LOOP TILL DONE
  
```

```

1241                                     ;CLEAR THE WORD COUNT REGISTER
1242 003536 104400 TST13: SCOPE
1243
1244 003540 012777 177777 175336      MOV    #177777, @R5WC      ;SET R5WC REGISTER EQUAL TO ALL ONES
1245 003546 005077 175332              CLR    @R5WC
1246 003552 005777 175326              TST    @R5WC              ;DID ALL BITS GET CLEARED
1247 003556 001401                      BEQ    +4                  ;YES
1248 003560 104010                      HLT    !WC                 ;NO
1249 003562 104400 TST14: SCOPE
1250
1251                                     ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1252
1253 003564 012777 177777 175316      MOV    #177777, @RSDA     ;SET ALL BITS
1254 003572 022777 177777 175310      CMP    #177777, @RSDA     ;ARE THE BITS SET
1255 003600 001401                      BEQ    +4                  ;YES
1256 003602 104004                      HLT    !DA                 ;NO
1257 003604 012777 125252 175276      MOV    #125252, @RSDA     ;SET THESE BITS
1258 003612 022777 125252 175270      CMP    #125252, @RSDA     ;ARE THEY =
1259 003620 001401                      BEQ    +4                  ;YES
1260 003622 104004                      HLT    !DA                 ;SHOULD BE 125252
1261 003624 012777 052525 175256      MOV    #52525, @RSDA      ;SET THESE BITS
1262 003632 022777 052525 175250      CMP    #52525, @RSDA      ;ARE THEY =
1263 003640 001401                      BEQ    +4                  ;YES
1264 003642 104004                      HLT    !DA                 ;SHOULD BE 52525
1265 003644 104400 TST15: SCOPE
1266
1267                                     ;FLOAT A 1 THROUGH RSDA
1268
1269 003646 012701 000001      FLOTDA: MOV    #1, GOOD      ;GET A 1
1270 003652 000241              CLC                          ;CLEAR CARRY
1271 003654 010177 175230      1$:    MOV    GOOD, @RSDA     ;FLOAT NUMBER
1272 003660 017700 175224      MOV    @RSDA, BAD           ;GET DA
1273 003664 020100              CMP    GOOD, BAD           ;COMPARE DA
1274 003666 001401              BEQ    +4                  ;DA CORRECT
1275 003670 104000              HLT                          ;BAD=DA GOOD=CORRECT ANS
1276 003672 006101              ROL                          ;ROTATE NUMBER
1277 003674 103367              BCC    1$                  ;LOOP TILL DONE
  
```

```

1278                                     ;CAN WE CLEAR THE RSDA REG.
1279 003676 104400 TST16: SCOPE
1280
1281 003700 012777 177777 175202      MOV      #177777, @RSDA ;SET RSDA TO ALL ONES
1282 003706 005077 175176              CLR      @RSDA          ;DID THEY SET
1283 003712 005777 175172              TST     @RSDA          ;TEST FOR ZERO RSDA
1284 003716 001401                      BEQ     +4              ;YES
1285 003720 104004                      HLT     !DA            ;ANS SHOULD BE 0
1286 003722 104400 TST17: SCOPE
1287
1288                                     ;SET AND CLEAR THE RSER REG.
1289
1290 003724 012777 177017 175162      MOV      #177017, @RSER ;SET THESE BITS
1291 003732 022777 177017 175154      CMP     #177017, @RSER ;DID THEY SET
1292 003740 001401                      BEQ     +4              ;YES
1293 003742 104002                      HLT     !ER            ;RSER SHOULD = 157017
1294 003744 112777 000001 175142      MOVB   #1, @RSER      ;A MOVB INST
1295 003752 022777 000001 175134      CMP     #1, @RSER     ;SHOULD MODIFY COMPLETE WD
1296 003760 001401                      BEQ     +4              ;OK
1297 003762 104002                      HLT     !ER
1298
1299 003764 104400 TST20: SCOPE
1300
1301 003766 012777 052005 175120      MOV      #52005, @RSER ;SET THESE BITS
1302 003774 022777 052005 175112      CMP     #52005, @RSER ;DID THEY SET
1303 004002 001401                      BEQ     +4              ;YES
1304 004004 104002                      HLT     !ER            ;ER SHOULD = 52005
1305 004006 104400 TST21: SCOPE
1306
1307 004010 012777 125012 175076      MOV      #125012, @RSER ;SET THESE BITS
1308 004016 022777 125012 175070      CMP     #125012, @RSER ;DID THEY SET
1309 004024 001401                      BEQ     +4              ;YES
1310 004026 104002                      HLT     !ER            ;ER SHOULD = 105012

```

```

1311 004030 104400          TST22: SCOPE
1312
1313 004032 012777 177017 175054          MOV    #177017,RSER    ;SET THESE BITS
1314 004040 005077 175050          CLR    RSER           ;CLEAR THEM
1315 004044 005777 175044          TST    RSER           ;DID THEY CLEAR
1316 004050 001401          BEQ    +4             ;YES
1317 004052 104002          HLT    !ER           ;SHOULD = 0
1318 004054 104400          TST23: SCOPE
1319
1320          ;SET AND CLEAR RSMR
1321
1322 004056 012777 000070 175040          MOV    #70,RSMR      ;SET THESE BITS
1323 004064 017767 175034 175122          MOV    RSMR,WORK    ;PUT INTO WORKABLE REG
1324 004072 042767 177700 175114          BIC    #177700,WORK  ;CLEAR JUNK
1325 004100 022767 000070 175106          CMP    #70,WORK     ;DID THEY SET
1326 004106 001401          BEQ    +4             ;YES
1327 004110 104220          HLT    !MR           ;SHOULD = 70
1328 004112 104400          TST24: SCOPE
1329
1330 004114 012777 000070 175002          MOV    #70,RSMR      ;SET BITS
1331 004122 005077 174776          CLR    RSMR          ;CLEAR THEM
1332 004126 032777 000077 174770          BIT    #77,RSMR     ;DID THEY CLEAR
1333 004134 001401          BEQ    +4             ;YES
1334 004136 104220          HLT    !MR           ;BITS (77) SHOULD = 0
1335 004140 104400          TST25: SCOPE
1336
1337 004142 012777 000050 174754          MOV    #50,RSMR      ;SET BITS
1338 004150 017767 174750 175036          MOV    RSMR,WORK    ;PUT IN WORKABLE REG
1339 004156 042767 177700 175030          BIC    #177700,WORK  ;CLEAR JUNK
1340 004164 022767 000050 175022          CMP    #50,WORK     ;DID THESE BITS SET
1341 004172 001401          BEQ    +4             ;YES
1342 004174 104220          HLT    !MR           ;BITS (50) SHOULD BE SET
1343 004176 104400          TST26: SCOPE
1344
1345 004200 012777 000020 174716          MOV    #20,RSMR      ;SET BITS
1346 004206 017767 174712 175000          MOV    RSMR,WORK    ;PUT INTO WORKABLE REG
1347 004214 042767 177700 174772          BIC    #177700,WORK  ;CLEAR JUNK
1348 004222 022767 000020 174764          CMP    #20,WORK     ;DID THEY SET
1349 004230 001401          BEQ    +4             ;YES
1350 004232 104220          HLT    !MR           ;MR SHOULD AT LEAST HAVE A (21)

```

F03

```

1351          ;*****
1352          ;TEST 27          TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1353          ;*****
1354 004234 104400 TST27: SCOPE
1355
1356 004236 104414 BITST: CLRDK          ;CLEAR ALL RS REG
1357 004240 012777 003566 174632 MOV          #3566,ARSCS1 ;LOAD CS1
1358 004246 112777 000005 174660 MOVB         #5,ARSCS1B ;LOAD BIT
1359 004254 022777 004766 174616 CMP          #4766,ARSCS1 ;DID IT LOAD?
1360 004262 001401 BEQ          .+4 ;YES
1361 004264 104001 HLT          !CS1
1362 004266 112777 000032 174604 MOVB         #32,ARSCS1
1363 004274 022777 004632 174576 CMP          #4632,ARSCS1
1364 004302 001401 BEQ          .+4
1365 004304 104001 HLT          !CS1 ;CS1 SHOULD = 4632
1366
1367 004306 104400 TST30: SCOPE
1368
1369 004310 016777 174644 174564 BITCS2: MOV        UNUM,ARSCS2 ;LOAD UNIT NUMBER
1370 004316 052777 177400 174556 BIS          #177400,ARSCS2 ;LOAD ALL BITS
1371 004324 105077 174606 CLRB        ARSCS2B ;CLR UPPER BYTE
1372 004330 016701 174624 MOV        UNUM,GOOD ;GET UNIT NO.
1373 004334 052701 000100 BIS          #100,GOOD ;SET OR BIT
1374 004340 017700 174536 MOV        ARSCS2,BAD ;GET CS2
1375 004344 020001 CMP        BAD,GOOD ;IS CS2 CORRECT?
1376 004346 001401 BEQ          .+4 ;YES
1377 004350 104000 HLT          ;LOAD BYTE DID NOT WORK
1378
1379 004352 104400 TST31: SCOPE
1380
1381 004354 012777 025252 174522 BITWC: MOV        #25252,ARSWC ;LOAD WC
1382 004362 112777 000377 174550 MOVB        #377,ARSWCB ;LOAD BIT
1383 004370 022777 177652 174506 CMP        #177652,ARSWC ;DID IT LOAD?
1384 004376 001401 BEQ          .+4 ;YES
1385 004400 104010 HLT        !WC ;NO WC SHOULD =177652
1386 004402 112777 000123 174474 MOVB        #123,ARSWC
1387 004410 022777 177523 174466 CMP        #177523,ARSWC
1388 004416 001401 BEQ          .+4
1389 004420 104010 HLT        !WC ;WC SHOULD = 177523
1390
1391 004422 104400 TST32: SCOPE
1392
1393 004424 012777 025252 174454 BITBA: MOV        #25252,ARSBAB ;LOAD DA
1394 004432 112777 000377 174502 MOVB        #377,ARSBAB ;LOAD BIT
1395 004440 022777 177652 174440 CMP        #177652,ARSBAB ;DID IT LOAD?
1396 004446 001401 BEQ          .+4 ;YES
1397 004450 104020 HLT        !BA ;DA SHOULD =177652
1398 004452 112777 000125 174426 MOVB        #125,ARSBAB
1399 004460 022777 177524 174420 CMP        #177524,ARSBAB
1400 004466 001401 BEQ          .+4
1401 004470 104020 HLT        !BA ;BA SHOULD = 177525
1402 004472 104414 CLRDK          ;CLEAR ALL RS REG

```


G03

```

1403 :*****
1404 :TEST 33 LOAD RSDB WITH ALL ONES AND ALL ZEROS
1405 :*****
1406 004474 104400 †TST33: SCOPE
1407 004476 104414 ZERONE: CLRDK
1408 004500 005077 174416 CLR †RSDB ;CLEAR ALL RS REG
1409 004504 012777 174410 174410 MOV †177777, †RSDB ;LOAD DB WITH ALL 0
1410 004512 012767 002000 174474 MOV †2000, WORK ;LOAD DB WITH ALL ONES
1411 004520 012701 000300 MOV †300, GOOD ;TIME OUT ROUTINE
1412 004524 056701 174430 BIS UNUM, GOOD ;GET CORRECT FOR CS2
1413 004530 017700 174346 2S: MOV †RSCS2, BAD ;GET CS2
1414 004534 020100 CMP GOOD, BAD ;IS IT CORRECT?
1415 004536 001404 BEQ 3S ;YES
1416 004540 005367 174450 DEC WORK ;TO WAIT FOR OR
1417 004544 001371 BNE 2S ;TO SET
1418 004546 104200 HLT !CS2 ;OR SHOULD BE SET
1419 004550 005001 3S: CLR GOOD ;LOAD BAD WITH DB
1420 004552 017700 174344 MOV †RSDB, BAD ;IS BAD CORRECT
1421 004556 020100 CMP GOOD, BAD ;YES
1422 004560 001401 BEQ .+4 ;COULD NOT FLOAT 0 THROUGH DB
1423 004562 104000 HLT ;LOAD GOOD WITH ANS
1424 004564 012701 177777 MOV †-1, GOOD ;GET DATA FROM DB
1425 004570 017700 174326 MOV †RSDB, BAD ;IS DB CORRECT
1426 004574 020100 CMP GOOD, BAD ;YES
1427 004576 001401 BEQ .+4 ;BAD SHOULD = 177777
1428 004600 104000 HLT

```

H03

```
1429 ;TEST INTERRUPT IN THE RH11
1430 ;BY MOVING 300 INTO RHCS1
1431 ;*****
1432 ;TEST 34 TEST INTERRUPT IN RH11
1433 ;*****
1434 004602 104400 TST34: SCOPE
1435 004604 104414 INT: CLDK
1436 004606 012777 004660 174314 MOV #PGTRAP,ARVVEC ;CLEAR ALL ERRORS
1437 004614 012777 000340 174310 MOV #340,ARVCPS ;SET UP VECTOR
1438 004622 012737 000200 177776 MOV #200,ARPS ;SET TRAP PS
1439 004630 012777 000300 174242 MOV #300,ARSCS1 ;SET PS AT PRIORITY 4
1440 004636 012767 000500 174350 MOV #500,WORK ;THIS SHOULD CAUSE A TRAP
1441 004644 005367 174344 1S: DEC WORK ;SETUP LOOP
1442 004650 001375 BNE 1S ;DEC LOOP SHOULD
1443 004652 104001 HLT !CS1 ;INTERRUPT BEFORE LOOP IS DONE
1444 004654 000167 000014 JMP INTDON ;SHOULD NEVER GET HERE
1445 004660 022626 PGTRAP: CMP (6)+,(6)+ ;GET OUT
1446 004662 022777 004200 174210 CMP #4200,ARSCS1 ;TRAP OK
1447 004670 001401 BEQ +4 ;DID IE CLEAR?
1448 004672 104001 HLT !CS1 ;YES
1449 004674 INTDON: ;IE SHOULD BE CLEARED
```

```

1450 :*****
1451 :TEST 35 MAINTENANCE TIMING TEST
1452 :*****
1453 004674 104400 TST35: SCOPE
1454
1455 :MODULE TESTED G092
1456 :THE FOLLOWING TEST ON THE RS04 DISK IS A SINGLE-STEPPED
1457 :MAINTENANCE MODE TEST ON THE RS04 TIMING LOGIC. THE ACTUAL
1458 :DISK SURFACE IS SUBSTITUTED BY THE MAINTENANCE RESISTER--I.E.
1459 :THE PROGRAM WILL SUPPLY ALL "DISK CLOCK" PULSES TO DRIVE THE
1460 :TIMING LOGIC. WE ARE TESTING THE ENTIRE TIMING TRACK LOGIC, INCLUDING, INDEX
1461 :PULSE FUNCTION, RESYNC AREA, SECTOR COUNTERS, ETC.
1462
1463 :PUT DRIVE IN MAINTENANCE MODE
1464 004676 104414 MRTIME: CLDK :CLEAR DRIVE REGISTERS
1465 004700 052767 001040 174260 BIS #1040,ONCEE :SET CLK CNT
1466 004706 104430 MRIND :SEND INDEX PULSE TO MR REG
1467 004710 104420 MRCK :CHECK MAINTENANCE REG FOR
1468 004712 022701 22701 :22701
1469 004714 104424 MRINT :INIT MAINT MODE (CLEAR MRSP)
1470 :BY SENDING 2 CLOCK PULSES
1471 004716 104430 MRIND :SEND MAINT INDEX PULSE
1472
1473 004720 104420 MRCK :CHECK MAINT REG TO
1474 004722 022701 22701 :EQUAL 22701
1475 004724 104000 HLT :MR=BAD GOOD=CORRECTIONS
1476 :COULD NOT INITIALIZE MR REG
1477 :INDEX PULSE SHOULD CLEAR LOOK-AHEAD REG
1478
1479 004726 005777 174166 TST 2RSLA :IS RSLA CLEARED
1480 004732 001401 BEQ +4 :YES
1481 004734 104224 HLT !MR!LA :RSLA SHOULD BE CLEARED
1482 :WITH THE INDEX PULSE
1483
1484 :PERFORM MAINTENANCE CLOCK OPERATION 1024 TIMES TO
1485 :PROVIDE CLOCK 10 STEP TIMING THRU RESYNC PERIOD.
1486 :IF SECTOR PULSE IS ASSERTED DURING THIS LOOP
1487 :CHECK SECTOR BOUNDARY COUNTER AND E12
1488
1489 004736 012767 001000 174236 MRTIM1: MOV #512.,REPT
1490 004744 104422 MRCLK :CLOCK MAINT REG WITH AN 11 AND A 1
1491 004746 104420 MRCK :CHECK MR REG TO
1492 004750 072701 72701 :EQUAL 72701
1493 004752 104000 HLT :MR = BAD, GOOD = CORRECT ANS
1494 004754 104422 MRCLK :CLOCK MR
1495 004756 104420 MRCK :CHECK MR TO
1496 004760 022701 22701 :EQUAL 22701
1497 004762 104000 HLT :BAD=MR REG GOOD=CORRECTIONS
1498 004764 005367 174212 DEC REPT :IS THE LOOP DONE YET?
1499 004770 001365 BNE MRTIM1 :NO-LOOP
    
```

J03

```

1500                                   ;ENTER ONE MORE CLOCK SECTOR PULSE SHOULD BE ASSERTED
1501                                   ;IF NOT, CHECK SECTOR BOUNDARY COUNTER, SECTOR BOUNDARY FF (E21) AND E12
1502
1503   004772   104422                   MRCLK                                   ;CLOCK MAINT REG WITH A 11 AND A 1
1504   004774   104420                   MRCK                                   ;CHECK MR REG TO
1505   004776   072301                   72301                               ;EQUAL 72301
1506   005000   104000                   HLT                                   ;MR=BAD GOOD=CORRECTIONS
1507   005002   104422                   MRCLK                               ;CLOCK MR WITH 11 AND A 1
1508   005004   104420                   MRCK                               ;CHECK MAINT REG
1509   005006   022301                   22301                               ;TO EQUAL 22301
1510   005010   104000                   HLT                               ;MR=BAD GOOD-CORRECT ANS
1511   005012   005777   174102           TST                               ;DOES LOOK AHEAD REG=0
1512   005016   001401                   BEQ                               ;YES-CONT
1513   005020   104224                   HLT                               ;LOOK AHEAD REG SHOULD=0
1514                                   ;PERFORM MAINTENANCE CLOCK OPERATION 80 TIMES TO PROVIDE
1515                                   ;CLOCK PULSES TO STEP THRU 1ST SECTOR PRE-AMBLE AREA
1516
1517   005022   005002                   MRT2:   CLR           R2                   ;CLEAR R2 FOR SECTOR COMPARE WITH LA REG
1518   005024   012767   000050   174150   MOV           #40.,REPT           ;80 CLOCKS TO STEP THRU PRE-AMBLE
1519   005032   104422                   MRT2A:  MRCLK                       ;CLOCK MR WITH A 11 AND A 1
1520   005034   104420                   MRCK                               ;CHECK MAINT REG
1521   005036   073701                   73701                               ;EQUAL 73701
1522   005040   104000                   HLT                               ;MR = BAD GOOD = CORRECT ANS
1523   005042   104422                   MRCLK                               ;CLOCK MR REG
1524   005044   104420                   MRCK                               ;CHECK MR REG
1525   005046   023701                   23701                               ;TO EQUAL 23701
1526   005050   104000                   HLT                               ;MR = BAD GOOD = CORRECTANS
1527   005052   005367   174124           DEC REPT                           ;REPEAT
1528   005056   001365                   BNE           MRT2A               ;LOOP 40 TIMES
1529
1530                                   ;SUPPLY CLOCKS TO STEP THROUGH THE DATA AREA IN THE SECTOR
1531   005060   012767   002200   174114   MOV           #9.*128.,REPT       ;18 CLOCKS PER DATA WORD
1532   005066   104422                   MRT2B:  MRCLK                       ;CLOCK MR WITH A 11 AND A 1
1533   005070   104420                   MRCK                               ;CHECK MAINT REG
1534   005072   073701                   73701                               ;TO EQUAL 73701
1535   005074   104000                   HLT                               ;MR = BRD GOOD = CORRECT ANS
1536   005076   104422                   MRCLK                               ;CLOCK MR REG
1537   005100   104420                   MRCK                               ;CHECK MR REG
1538   005102   023701                   23701                               ;TO EQUAL 23701
1539   005104   104000                   HLT                               ;MR=BAD GOOD=CORRECTANS
1540   005106   005367   174070           DEC           REPT               ;REPEAT
1541   005112   001365                   BNE           MRT2B               ;LOOP
  
```

K03

```

1542      ; SUPPLY ENOUGH MAINT CLOCKS TO STEP THROUGH THE CRC AREA
1543      ; AND THE DEAD BAND ON THE SECTOR
1544
1545 005114 012767 000214 174060      MOV      #140.,REPT      ; AMOUNT OF CLOCKS TO END OF SECTOR
1546 005122 104422      MRT2C: MRCLK      ; CLOCK MR WITH A 11 AND A 1
1547 005124 104420      MRCK      ; CHECK MAINT REG
1548 005126 073701      73701      ; TO EQUAL 73701
1549 005130 104000      HLT      ; MR = BAD GOOD = CORRECT ANS
1550 005132 104422      MRCLK      ; CLOCK MR REG
1551 005134 104420      MRCK      ; CHECK MAINT REG
1552 005136 023701      23701      ; TO EQUAL 23701
1553 005140 104000      HLT      ; MR=BAD GOOD=CORRECT ANS
1554 005142 005367 174034      DEC      REPT      ; REPEAT
1555 005146 001365      BNE      MRT2C      ; LOOP
1556 005150 104422      MRCLK      ; CLOCK MR REG
1557 005152 104420      MRCK      ; CHECK MR REG
1558 005154 073701      73701      ; TO EQUAL 73701
1559 005156 104000      HLT      ; MR = BAD GOOD = CORRECT ANS
1560      ; ONE MORE CLOCK SHOULD CAUSE SECTOR PULSE
1561      ; IF NOT, CHECK E16-6
1562
1563 005160 104422      MRCLK      ; CLOCK MR WITH A 11 AND A 1
1564 005162 104420      MRCK      ; MAINT REG SHOULD
1565 005164 023701      23701      ; EQUAL 22301
1566 005166 104000      HLT      ; MR=BAD GOOD=CORRECT ANS
1567 005170 104422      MRCLK      ; CLOCK MR WITH A 11 AND A 1
1568 005172 104420      MRCK      ; MAINT REG
1569 005174 072301      72301      ; SHOULD EQUAL 72301
1570 005176 104000      HLT      ; MR=BAD GOOD=CORRECT ANS
1571
1572      ; LOOK-AHEAD REGISTER SHOULD NOW POINT TO SECTOR 1 (OR 4000 IF INTERLEAVED)
1573
1574 005200 022777 000002 173720      CMP      #2,ARSOT      ; INTERLEAVED?
1575 005206 001403      BEQ      3$          ; NO
1576 005210 062702 004000      ADD      #4000,R2      ; YES
1577 005214 000402      BR       2$          ; CONT
1578 005216 062702 000100 3$:  ADD      #100,R2      ; INCREMENT SECTOR COMPARE
1579 005222 020277 173672 2$:  CMP      R2,ARSLA     ; LA REG SHOULD=100
1580 005226 001401      BEQ      1$          ; LA IS CORRECT
1581 005230 104224      HLT      !MR!LA      ; LA SHOULD=100
  
```

L03

MAINDEC-11-DERSD-C RH70-RSD4 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 38
 DERSDC.SRC 17-MAY-77 14:16 TST35 MAINTENANCE TIMING TEST

```

1582
1583
1584
1585
1586
1587 005232 012767 000076 173744 1S:  MOV  #62.,REPT1
1588 005240 012767 005152 173734 MRT3: MOV  #2666.,REPT
1589 005246 104422 3S:  MRCLK
1590 005250 005367 173726  DEC  REPT
1591 005254 001374  BNE  3S
1592 005256 104422  MRCLK
1593 005260 104420  MRCK
1594 005262 022701 22701
1595 005264 104000  HLT
1596 005266 104422  MRCLK
1597 005270 104420  MRCK
1598 005272 072301 72301
1599 005274 104000  HLT
1600 005276 022777 000002 173622  CMP  #2,RSDT
1601 005304 001420  BEQ  6S
1602 005306 032767 001000 173652  BIT  #BIT9,ONCEE
1603 005314 001406  BEQ  4S
1604 005316 042767 001000 173642  BIC  #BIT9,ONCEE
1605 005324 162702 004000  SUB  #4000,R2
1606 005330 000406  BR   6S
1607 005332 052767 001000 173626 4S:  BIS  #BIT9,ONCEE
1608 005340 062702 004000  ADD  #4000,R2
1609 005344 000402  BR   5S
1610 005346 062702 000100 6S:  ADD  #100,R2
1611 005352 017700 173542 5S:  MOV  @RSLA,BAD
1612 005356 010201  MOV  R2,GOOD
1613 005360 020100  CMP  GOOD,BAD
1614 005362 001401  BEQ  1S
1615 005364 104000  HLT
1616
1617 005366 005367 173612 1S:  DEC  REPT1
1618 005372 001322  MRT3
1619 005374 012767 005152 173600 2S:  MOV  #2666.,REPT
1620 005402 104422  MRCLK
1621 005404 005367 173572  DEC  REPT
1622 005410 001374  BNE  2S
1623 005412 017700 173502  MOV  @RSLA,BAD
1624 005416 012701 007777  MOV  #7777,GOOD
1625 005422 020100  CMP  GOOD,BAD
1626 005424 001401  BEQ  .+4
1627 005426 104000  HLT

```

```

:REPEAT NEXT STEPS 62 TIMES. LOOK-AHEAD REGISTER SHOULD INCREMENT
:TO SHOW NEXT SECTOR. CHECKS FOR ALL SECTORS. IF DRIVE IS NOT
:INTERLEAVED, LA = 200,300, ETC. IF DRIVE IS INTERLEAVED,
:LA = 100, 4100, 200, 4200 ETC. SEE SERVICE MANUAL FOR DETAILS.

```

```

: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

```

M03

```

1628 :*****
1629 :TEST 36 SECTOR FRACTION TEST
1630 :*****
1631 005430 104400 TST36: SCOPE
1632 :MODULE TESTED G092
1633 :CLOCK THROUGH AN ENTIRE TRACK IN MAINT MODE WHILE
1634 :CHECKING FOR THE PROPER OPERATION OF THE LOOK-AHEAD REGISTER AND
1635 :THE SECTOR FRACTION COUNTER. WHEN THE LAST WORD IS BEING TRANSFERRED,
1636 :SECTOR AND FRACTION IS EQUAL TO 7777 TO INDICATE LAST WORD ON THIS TRACK --
1637 :HANDLE END OF TRACK SPECIAL FOR THE LOOK-AHEAD REGISTER WILL CLEAR THE
1638 :FRACTION BITS IF ANOTHER WORD IS CLOCKED. RSLA SHOULD INDICATE 7700 ON
1639 :ANOTHER MAINTENANCE CLOCK.
1640
1641 005432 104414 MRT4: CLDK ;CLEAR DRIVE REGISTERS
1642 005434 052767 000040 173524 BIS #40,ONCEE ;SET FLAG BITS
1643 005442 042767 003000 173516 BIC #3000,ONCEE
1644 005450 005067 173520 CLR MCCNT ;CLEAR MAINT CLOCK COUNTER
1645 005454 005002 CLR R2 ;CLEAR R2 FOR SECTOR COUNTER
1646 005456 104430 MRIND ;SEND INDEX PULSE TO MR REG
1647 005460 104420 MRCK ;CHECK MR REG
1648 005462 022701 22701 ;TO EQUAL 22701
1649 005464 104424 MRINT ;INIT MAINT MODE
1650 005466 104430 MRIND ;ISSUE A MAINT INDEX PULSE
1651 ;TO CLEAR THE DRIVE
1652 005470 104420 MRCK ;CHECK MAINT REG
1653 005472 022701 22701 ;TO EQUAL 22701
1654 005474 104000 HLT ;MR=BAD GOOD=CORRECT ANS
1655
1656 ;ISSUE 1024 MAINT CLOCKS TO STEP THROUGH THE RESYNC AREA
1657
1658 005476 012767 001000 173476 MRT4A: MOV #512.,REPT ;COUNT TO STEP THRU RESYNC AREA
1659 005504 104422 MRCLK ;CLOCK THROUGH RESYNC
1660 005506 104420 MRCK ;CHECK MAINT REG
1661 005510 072701 72701 ;TO EQUAL 72701
1662 005512 104000 HLT ;MR = BAD GOOD = CORRECT ANS
1663 005514 005777 173400 TST @RSLA ;IS RSLA=TO 0
1664 005520 001401 BEQ .+4 ;YES
1665 005522 104204 HLT !LA ;RSLA SHOULD=0 DURING RESPONSE
1666 005524 104422 MRCLK ;CLOCK MR REG
1667 005526 104420 MRCK ;CHECK MR REG
1668 005530 022701 22701 ;TO EQUAL 22701
1669 005532 104000 HLT ;BAD=MR GOOD=CORRECT ANS
1670 005534 005777 173360 TST @RSLA ;IS RSLA=TO 0
1671 005540 001401 BEQ .+4 ;YES
1672 005542 104204 HLT !LA ;RSLA SHOULD=0 DURING RESPONSE
1673 005544 005367 173432 DEC REPT ;LOOP THROUGH
1674 005550 001355 BNE MRT4A ;RESYNC AREA
1675
1676 ;ONE MORE PULSE SHOULD CAUSE THE FIRST SECTOR PULSE
1677
1678 005552 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1679 005554 104420 MRCK ;CHECK MAINT REG FOR SECTOR PULSE
1680 005556 072301 72301 ;MR SHOULD=72301
1681 005560 104000 HLT ;MR=BAD GOOD=CORRECT ANS

```

```

1682 005562 104422      MRT4B: MRCLK      ;CLOCK MR REG WITH A 11 AND A 1
1683 005564 104420      MRCK      ;CHECK MAINT REG
1684 005566 022301      22301     ;TO EQUAL 22301
1685 005570 104000      HLT      ;MR=BAD GOOD=CORRECT ANS
1686
1687      ;SECTOR FRACTION BITS IN LOOK-AHEAD REGISTER SHOULD BE CLEARED (EQUAL TO 00)
1688
1689 005572 017700 173322      MOV      2RSLA,BAD ;GET RSLA
1690 005576 010201      MOV      R2,GOOD   ;GET CORRECT ANS
1691 005600 020100      CMP      GOOD,BAD  ;IS THE RSLA REG CORRECT
1692 005602 001401      BEQ      1$        ;YES
1693 005604 104000      HLT      ;RSLA=BAD GOOD=CORRECTANS
1694
1695      ;STEP THROUGH THE PREAMBLE AREA AND SECTOR DATA
1696      ;AREA WHILE CHECKING THE SECTOR FRACTION
1697
1698 005606 012767 000244 173366 1$: MOV      8164.,REPT ;FOR FIRST FRACTION CHANGE
1699 005614 104422      MRT4C: MRCLK      ;CLOCK MR REG WITH A 11 AND A 1
1700 005616 017700 173276      MOV      2RSLA,BAD ;GET RSLA
1701 005622 010201      MOV      R2,GOOD   ;GET CORRECT ANS
1702 005624 020001      CMP      BAD,GOOD  ;IS RSLA CORRECT
1703 005626 001401      BEQ      1$        ;YES
1704 005630 104000      HLT      ;BAD=RSLA GOOD=CORRECT ANS
1705 005632 005367 173344 1$: DEC      REPT    ;LOOP ON
1706 005636 001366      BNE      MRT4C    ;PREAMBLE AREA
1707
1708      ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1709
1710 005640 104422      MRCLK      ;CLOCK MR WITH A 11 AND A 1
1711 005642 005202      INC      R2        ;COUNT THE FRACTION
1712 005644 017700 173250      MOV      2RSLA,BAD ;GET RSLA
1713 005650 010201      MOV      R2,GOOD   ;GET CORRECT ANS
1714 005652 020001      CMP      BAD,GOOD  ;IS RSLA CORRECT?
1715 005654 001401      BEQ      2$        ;YES
1716 005656 104000      HLT      ;RSLA=BAD GOOD=CORRECT ANS
1717
1718      ;FIRST FRACTION CHANGES AFTER 164 MAINT. CLKS, THE REST
1719      ;CHANGE AFTER 40 MAINTENANCE CLOCKS
1720
1721 005660 012767 000076 173314 2$: MOV      862.,REPT ;COUNT FOR WORDS IN A SECTOR
1722 005666 012767 000047 173310 MRT4D: MOV      839.,REPT1 ;COUNT FOR SECT FRACT TO CHANGE
1723 005674 104422      MRT4E: MRCLK      ;CLOCK MR WITH A 11 AND A 1
1724 005676 017700 173216      MOV      2RSLA,BAD ;GET RSLA
1725 005702 010201      MOV      R2,GOOD   ;GET CORRECT ANS
1726 005704 020100      CMP      GOOD,BAD  ;IS RSLA CORRECT?
1727 005706 001401      BEQ      1$        ;YES
1728 005710 104000      HLT      ;RSLA=BAD GOOD=CORRECT ANS
1729 005712 005367 173266 1$: DEC      REPT1   ;LOOP
1730 005716 001366      BNE      MRT4E
  
```



```

1731 ;ONE MORE CLOCK TO CAUSE THE SECTOR FRACTION TO CHANGE
1732
1733 005720 104422 MRCLK ;CLOCK MR WITH A 11 AND A 1
1734 005722 022702 007777 CMP #7777,R2 ;AT THE LAST SECTOR-LAST FRACTION?
1735 005726 001472 BEQ MRT4F ;YES, FINISH THE SECTOR
1736 005730 005202 INC R2 ;NO, ADD 1 TO FRACTION
1737 005732 017700 173162 4S: MOV #RSLA,BAD ;GET RSLA
1738 005736 022777 000002 173162 CMP #2,RSDT ;IS THIS DIRVE INTERLEAVED?
1739 005744 001431 BEQ 12S ;NO
1740 005746 032767 002000 173212 BIT #BIT10,ONCEE ;HAS REPT GONE TO ZERO YET FOR THIS SECTOR?
1741 005754 001425 BEQ 12S ;NO
1742 ;RSLA NOW POINTS TO NEXT INTERLEAVED SECTOR BIT 9 IN ONCEE
1743 ;INDICATES WHETHER RSLA SHOULD NOW BE BETWEEN
1744 ;0000-3700(1) OR 4000-7700(0).
1745 005756 032767 001000 173202 BIT #BIT9,ONCEE ;SHOULD RSLA BE BETWEEN 0-3700?
1746 005764 001004 BNE 9S ;YES
1747 005766 052767 001000 173172 BIS #BIT9,ONCEE ;SET FOR NEXT PASS
1748 005774 000406 BR 10S
1749 005776 042767 001000 173162 9S: BIC #BIT9,ONCEE ;CLEAR FOR NEXT PASS
1750 006004 042702 004000 BIC #4000,R2 ;MAKE EXPECTED RSLA LESS THAN 4000
1751 006010 000404 BR 5S
1752 006012 062702 004000 10S: ADD #4000,R2 ;COMPENSATE FOR INTERLEAVING
1753 006016 162702 000100 SUB #100,R2
1754 006022 042767 002000 173136 5S: BIC #BIT10,ONCEE ;CLEAR FLAG FOR NEXT SECTOR
1755 006030 010201 12S: MOV R2,GOOD ;GET CORRECT ANSWER FOR RSLA
1756 006032 020100 CMP GOOD,BAD ;IS RSLA CORRECT
1757 006034 001401 BEQ 2S ;YES
1758 006036 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS
1759 006040 005367 173136 2S: DEC REPT ;HAS SECTOR FRACTION REACHED 77?
1760 006044 001310 BNE MRT4D ;NO
1761
1762 ;CHECK FOR END OF ONE SECTOR OR BEGINNING OF NEXT
1763
1764 006046 010203 11S: MOV R2,R3
1765 006050 042703 177700 BIC #177700,R3 ;CHECK SECTOR FRACTION
1766 006054 022703 000077 CMP #77,R3 ;END OF SECTOR?
1767 006060 001402 BEQ 3S ;YES
1768 006062 000167 177474 JMP MRT4B ;NO, BEGINNING OF NEXT
1769 006066 012767 000025 173110 3S: MOV #21,REPT1 ;SETUP LOOP TO FINISH
1770 006074 012767 000001 173100 MOV #1,REPT ;THIS SECTOR
1771 006102 052767 002000 173056 BIS #BIT10,ONCEE ;REPT HAS GONE TO ZERO FOR THIS SECTOR
1772 006110 000167 177560 JMP MRT4E ;LOOP
1773
1774 006114 012767 000021 173060 MRT4F: MOV #17,REPT
1775 006122 104422 1S: MRCLK ;CLOCK MR WITH A 11 AND A 1
1776 006124 017700 172770 MOV #RSLA,BAD ;GET RSLA
1777 006130 010201 MOV R2,GOOD ;R2 SHOULD=7777
1778 006132 020100 CMP GOOD,BAD ;IS RSLA CORRECT-END OF DISK?
1779 006134 001401 BEQ 2S ;YES
1780 006136 104000 HLT ;RSLA=BAD GOOD=CORRECT ANS (7777)
1781 006140 005367 173036 2S: DEC REPT ;FINISH
1782 006144 001366 BNE 1S ;LOOP
    
```

C04

1783
1784
1785
1786 006146 104422
1787 006150 017700 172744
1788 006154 012701 007700
1789 006160 020100
1790 006162 001401
1791 006164 104000
1792 006166 104430
1793
1794 006170 017700 172724
1795 006174 005001
1796 006176 020100
1797 006200 001401
1798 006202 104000
1799 006204 104420
1800 006206 022701
1801 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 87700,GOOD :GET CORRECT ANS
CMP GOOD,BAD :IS RSLA CORRECT?
BEQ 1\$:YES
HLT :RSLA=BAD GOOD=CORRECT ANS
1\$: 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 2\$:YES
HLT :RSLA=BAD GOOD=CORRECT ANS
2\$: MRCK :CHECK MR REG
22701 :TO EQUAL 22701
HLT :MR=BAD GOOD=CORRECT ANS

```

1802 :*****
1803 :TEST 37 ILLEGAL FUNCTION TEST
1804 :*****
1805 006212 104400 TST37: SCOPE
1806
1807 :MODULE TESTED M7759, M7770
1808 :TEST ILLEGAL FUNCTION (ILF) IN RSER. SEND AN ILLEGAL
1809 :FUNCTION CODE TO THE DRIVE CONTROL REGISTER WITHOUT SETTING
1810 :THE GO BIT. THE "ILF" BIT SHOULD NOT BE SET. THE "GO" BIT
1811 :IS THEN SET. A CHECK IS THEN MADE FOR "ATA" AND "ERR" TO BE
1812 :SET IN THE DRIVE STATUS REGISTER (RSDS) AND "ILF" IN THE
1813 :DRIVE ERROR REGISTER (RSER). ALL ILLEGAL FUNCTION CODES ARE
1814 :CHECKED.
1815 :ILLEGAL FUNCTIONS ARE DETECTED ON M7759 BY E20-8
1816
1817 006214 104414 MRILF: CLRDK ;CLEAR ALL THE DRIVE REGISTERS
1818 006216 042767 000040 172742 BIC #BITS,ONCEE ;CLEAR CLOCK CNT FLAG
1819 006224 032767 000002 172734 BIT #BIT1,ONCEE ;WAS THERE AN ERROR
1820 006232 001002 BNE MRLF1 ;YES DO NOT CHANGE "ILF" CODE
1821 006234 012702 000003 MOV #3,R2 ;SETUP FIRST "ILF" CODE
1822 ;PUT DRIVE IN MAINTENANCE MODE
1823
1824 006240 104416 MRLF1: MRDMD ;PUT DRIVE INTO MAINT MODE
1825 006242 104420 MRCK ;CHECK MR REG TO
1826 006244 022701 22701 ;EQUAL 22701
1827 006246 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1828
1829 ;ASSERT A MAINTENANCE MODE DISK "INDEX" PULSE
1830
1831 006250 104430 MRLF2: MRIND
1832 006252 010277 172622 MOV R2,RSCS1 ;SEND "ILF" WITH THE "GO" BIT
1833 006256 017700 172630 MOV RSDS,BAD ;GET DRIVE STATUS REG
1834 006262 012701 150600 MOV #150600,GOOD ;GET CORRECT ANS
1835 006266 020100 CMP GOOD,BAD ;IS RSDS CORRECT?
1836 006270 001440 BEQ IS ;YES
1837 006272 104402 006276 TYPE .+2 ;.ASCIZ <15><12>"ILLEGAL FUNCTION CODE SENT TO DRIVE="
1838 006346 010267 172642 MOV R2,WORK ;GET FUNCTION CODE
1839 006352 016746 172636 MOV WORK,-(6) ;PUT WORK ON STACK
1840 006356 104406 TYPES ;TYPE STACK IN OCTAL - SUPPRESS
1841 006360 052767 000002 172600 BIS #BIT1,ONCEE ;SET ERROR BIT SO ILLEGAL FUN DOESN'T CHANGE
1842 006366 104000 HLT ;RSDS=BAD GOOD=CORRECT ANS
1843 006370 104040 HLT !DS
1844
1845 006372 042767 000002 172566 1$: BIC #BIT1,ONCEE ;CLEAR ERROR FLAG
1846 006400 017700 172510 MOV RRSER,BAD ;GET RSER
1847 006404 012701 000001 MOV #1,GOOD ;GET CORRECT ANS
1848 006410 020100 CMP GOOD,BAD ;DID "ILF" SET IN RSER
1849 006412 001404 BEQ 2$ ;YES
1850 006414 052767 000002 172544 BIS #BIT1,ONCEE ;SET ERROR BIT
1851 006422 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
1852 006424 042767 000002 172534 2$: BIC #BIT1,ONCEE ;CLEAR ERROR FLAG
    
```

```

1853          ;CLEAR THE DRIVE FOR THE NEXT "ILF" CODE PASS
1854 006432 104414          MRCILF: CLDK          ;CLEAR ERRORS
1855 006434 017700 172452  MOV      2RSDS,BAD      ;GET RSDS REG
1856 006440 012701 010600  MOV      #10600,GOOD  ;GET CORRECT ANS
1857 006444 020100          CMP      GOOD,BAD      ;DID "ATA" AND "ERR" CLEAR IN RSDS?
1858 006446 001435          BEQ      IS          ;YES
1859 006450 104402 006454  TYPE      ,.+2          ;.ASCIZ <15><12>"ATA AND ERR IN RSDS SHOULD CLEAR WITH I
1860 006532 052767 000002 172426  BIS      #BIT1,ONCEE
1861 006540 104000          HLT
1862 006542 042767 000002 172416 15:  BIC      #BIT1,ONCEE  ;RSDS=BAD GOOD=CORRECT ANS
1863 006550 017700 172340  MOV      2RSER,BAD      ;CLEAR ERROR FLAG
1864 006554 005001          CLR      GOOD          ;GET RSER
1865 006556 020100          CMP      GOOD,BAD      ;GET CORRECT ANS
1866 006560 001431          BEQ      2S          ;DID ILF CLEAR IN RSER
1867 006562 052767 000002 172376  BIS      #BIT1,ONCEE  ;YES
1868 006570 104402 006574  TYPE      ,.+2          ;SET ERROR BIT
1869 006642 104000          HLT          ;.ASCIZ <15><12>"ILF IN RSER SHOULD CLEAR WITH INIT"
1870 006644 042767 000002 172314 2S:  BIC      #BIT1,ONCEE  ;RSER=BAD GOOD=CORRECT ANS
1871          ;GET NEXT ILLEGAL FUNCTION COE ;CLEAR ERROR BIT
1872
1873 006652 062702 000002  MRLF3:  ADD      #2,R2          ;UPDATE ILF
1874 006656 022702 000011  CMP      #11,R2        ;IS THIS A ILF CODE
1875 006662 001773          BEQ      MRLF3        ;NO-UPDATE IT
1876 006664 022702 000021  CMP      #21,R2
1877 006670 001770          BEQ      MRLF3
1878 006672 022702 000031  CMP      #31,R2
1879 006676 001765          BEQ      MRLF3
1880 006700 022702 000051  CMP      #51,R2
1881 006704 001762          BEQ      MRLF3
1882 006706 022702 000061  CMP      #61,R2
1883 006712 001757          BEQ      MRLF3
1884 006714 022702 000071  CMP      #71,R2
1885 006720 001754          BEQ      MRLF3
1886 006722 022702 000101  CMP      #101,R2
1887 006726 001402          BEQ      ILFDON
1888 006730 000167 177304  JMP      MRLF1
1889 006734          ILFDON:

```

;FINISHED ALL ILF CODES GET OUT
;START NEXT ILF FUNCTION

F04

MAINDEC-11-DERSD-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 45
DERSDC.SRC 17-MAY-77 14:16 TST40 TEST NO-OP CODES 1 AND 21

```
1890 ;*****
1891 ;TEST 40 TEST NO-OP CODES 1 AND 21
1892 ;*****
1893 006734 104400 TST40: SCOPE
1894
1895 ;MODULE TESTED M7759
1896 006736 104414 MR0P: CLDK ;CLEAR ALL DRIVE REGISTERS
1897 006740 042767 000004 172220 BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
1898 006746 104416 MR0MD ;PUT DRIVE INTO MAINT MODE
1899 006750 104420 MRCK ;CHECK MR REG TO
1900 006752 022701 22701 ;EQUAL 22701
1901 006754 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
1902 ;SEND INDEX PULSE
1903 006756 032767 000010 172202 BIT #BIT3,ONCEE ;TESTING CODE I
1904 006764 001031 BNE 3$ ;NO CODE 21
1905 006766 012777 000001 172104 MOV #1,RSRCS1 ;LOAD NO-OP FUNCTION
1906 006774 012767 000001 172212 MOV #1,WORK ;LOAD NO-OP FUNCTION
1907 007002 005777 172106 TST RSR ;ANY ERRORS
1908 007006 001403 BEQ 1$ ;NO
1909 007010 004767 012254 JSR PC,NOPERR ;TYPE IT
1910 007014 104040 HLT !D$ ;TYPE ERROR
1911 007016 022777 010600 172066 1$: CMP #10600,RSRDS ;IS RSDS CORRECT
1912 007024 001403 BEQ 2$ ;YES
1913 007026 004767 012236 JSR PC,NOPERR ;RSDS SHOULD
1914 007032 104040 HLT !D$ ;EQUAL 10600
1915 007034 042767 000004 172124 2$: BIC #BIT2,ONCEE ;CLEAR ERROR FLAG
1916
1917 ;TEST NO-OP FUNCTION CODE 21
1918
1919 007042 052767 000010 172116 BIS #BIT3,ONCEE ;TEST TESTING CODE 21 FLAG
1920 007050 012767 000021 172136 3$: MOV #21,WORK ;LOAD CODE 21
1921 007056 012777 000021 172014 MOV #21,RSRCS1 ;LOAD FUNCTION
1922 007064 005777 172024 TST RSR ;ANY ERRORS?
1923 007070 001403 BEQ 4$ ;NO
1924 007072 004767 012172 JSR PC,NOPERR ;YES, TYPE ERROR
1925 007076 104040 HLT !D$ ;ERROR DURING NO-OP FUNCTION
1926 007100 022777 010600 172004 4$: CMP #10600,RSRDS ;IS RSDS CORRECT
1927 007106 001403 BEQ 5$ ;YES
1928 007110 004767 012154 JSR PC,NOPERR ;TYPE ERROR
1929 007114 104040 HLT !D$ ;RSDS SHOULD=10600
1930 007116 042767 000014 172042 5$: BIC #14,ONCEE ;CLEAR TEST BITS
```

G04

```

1931 :*****
1932 :TEST 41 TEST NO-OP FUNCTION WITH ERROR BITS SET
1933 :*****
1934 007124 104400 TST41: SCOPE
1935
1936 :MODULE TESTED M7759
1937 007126 104414 MROPER: CLRDK :CLEAR ALL REGISTERS
1938 007130 104416 MRDMD :PUT DRIVE INTO MAINT MODE
1939 007132 104420 MRCK :CHECK MR REG
1940 007134 022701 22701 :TO EQUAL 22701
1941 007136 104424 MRINT :INIT MAINT MODE (CLEAR MRSP)
1942 007140 104430 MRIND :SEND INDEX PULSE
1943
1944 007142 012777 177777 171744 MOV #-1,RSER :LOAD RSER WITH ERRORS
1945 007150 116701 172010 MOV UNCMP,GOOD :GET DRIVE UNDER TEST
1946 007154 017700 171736 MOV RSAS,BAD :GET RSAS REG
1947 007160 020100 CMP GOOD,BAD :DID ATA BIT SET CAUSED BY ERROR
1948 007162 001427 BEQ IS :YES
1949 007164 104402 007170 TYPE ,.+2 :.ASCIZ <15><12>"SET ERRORS IN RSER-RSAS IS INCORRECT"
1950 007240 104000 HLT :RSAS=BAD GOOD=CORRECT ANS
1951 007242 012767 000001 171744 1S: MOV #1,WORK :SETUP FOR NO-OP CODE 1
1952 007250 032767 000010 171710 BIT #BIT3,ONCEE :TESTING CODE 21?
1953 007256 001004 BNE 2S :YES
1954 007260 012777 000001 171612 MOV #1,RSRCS1 :SEND NO-OP CODE 1
1955 007266 000406 BR 3S :CHECK FOR ERRORS
1956 007270 012767 000021 171716 2S: MOV #21,WORK :SETUP FOR CODE 21
1957 007276 012777 000021 171574 MOV #21,RSRCS1 :SENT NO-OP CODE 21
1958 007304 017700 171604 3S: MOV RSER,BAD :GET RSER REG
1959 007310 012701 177017 MOV #177017,GOOD :GET CORRECT ANS
1960 007314 020100 CMP GOOD,BAD :DID RSER CHANGE WITH NO-OP
1961 007316 001411 BEQ 4S :NO
1962 007320 104402 007324 TYPE ,.+2 :.ASCIZ <15><12>"RSER "
1963 007334 004767 012024 JSR PC,CHG
1964 007340 104000 HLT :RSER=BAD GOOD=CORRECT ANS
1965 007342 017700 171550 4S: MOV RSAS,BAD :GET RSAS
1966 007346 116701 171612 MOV UNCMP,GOOD :GET CORRECT ANS
1967 007352 020100 CMP GOOD,BAD :IS RSAS CORRECT
1968 007354 001411 BEQ 5S :YES
1969 007356 104402 007362 TYPE ,.+2 :.ASCIZ <15><12>"RSAS "
1970 007372 004767 011766 JSR PC,CHG :TYPE ERROR
1971 007376 104000 HLT :RSAS=BAD GOOD=CORRECT ANS
1972 007400 017700 171506 5S: MOV RSDS,BAD :GET RSDS
1973 007404 012701 150600 MOV #150600,GOOD :GET CORRECT ANS
1974 007410 020100 CMP GOOD,BAD :DID RSDS CHANGE
1975 007412 001411 BEQ 6S :NO
1976 007414 104402 007420 TYPE ,.+2 :.ASCIZ <15><12>"RSDS "
1977 007430 004767 011730 JSR PC,CHG :TYPE ERROR
1978 007434 104000 HLT :RSDS=BAD GOOD=CORRECT ANS
1979 007436 032767 000010 171522 6S: BIT #BIT3,ONCEE :TESTING CODE 21
1980 007444 001005 BNE 7S :YES, GET OUT
1981 007446 052767 000010 171512 BIS #BIT3,ONCEE :SET CODE 21 FLAG
1982 007454 000167 177446 JMP MROPER :TEST CODE 21
1983 007460 042767 000010 171500 7S: BIC #BIT3,ONCEE :DONE CLEAR FLAG AND CONT.

```

H04

```

1984 :*****
1985 :TEST 42          BLOCK SEARCH TEST 1
1986 :*****
1987 007466 104400 TST42: SCOPE
1988
1989 :MODULE TESTED: M7759, M7754, M7771, M7770
1990 :A DRIVE SEARCH FUNCTION IS GIVEN TO THE DRIVE FOR SECTOR 3.
1991 : (SECTOR 41 IF SECTOR INTERLEAVING IS ENABLED) THE
1992 : POSITIONING IN PROGRESS BIT (PIP) AND THE DRIVE READY BIT
1993 : (DRY) IN THE DRIVE STATUS REGISTER (RSDS) ARE CHECKED. THE
1994 : ADDRESS CONFIRM BIT (AC) IS ALSO CHECKED.
1995
1996 007470 104414 MRSRCH: CLRDK      ;CLEAR ALL REGISTERS
1997 007472 052767 000040 171466 BIS          #BITS,ONCEE ;SET CLOCK FLAG
1998 007500 104416 MRDMD        ;PUT DRIVE INTO MAINTENANCE MOE
1999 007502 104420 MRCK          ;CHECK MR REG
2000 007504 022701 22701        ;TO EQUAL 22701
2001 007506 104424 MRINT        ;INIT MR REG (CLEAR MRSP)
2002 007510 104430 MRIND        ;CLOCK INDEX PULSE IN RSMR
2003 007512 012777 000003 171370 MOV          #3,RSDA  ;DO A SEARCH FOR SECTOR 3 OR 41
2004 007520 022777 000002 171400 CMP          #2,RSOT  ;INTERLEAVED?
2005 007526 001403 BEQ          4$      ;NO SECTOR 3
2006 007530 012777 000041 171352 MOV          #41,RSDA ;YES SECTOR 41
2007 007536 012777 000031 171334 4$: MOV        #31,RSCSI ;LOAD SEARCH COMMAND (M7759)
2008 007544 104426 DSK          ;CHECK RSDS
2009 007546 030400 30400      ;TO EQUAL 30400
2010 007550 104000 HLT          ;PIP SHOULD BE SET AND DRY SHOULD
2011                                     ;BE 0 FOR A DRIVE SEARCH CMD
2012 007552 012767 021506 171422 1$: MOV        #21506,REPT ;STEP THROUGH 3 SECTORS
2013 007560 104422 MRCLK        ;CLOCK MR
2014 007562 104426 DSK          ;RSDS SHOULD NOT
2015 007564 030400 30400      ;CHANGE TILL CLOCKING IS COMPLETED
2016 007566 104000 HLT          ;TO REACH SECTOR 3
2017 007570 005367 171406 DEC          REPT      ;KEEP CLOCKING TILL
2018 007574 001371 BNE          1$      ;SECTOR 3 HAS BEEN REACHED
2019 ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2020 007576 104422 MRCLK        ;CLOCK MR REG
2021 007600 104426 DSK          ;CHECK FOR "ATA" AND "DRY"
2022 007602 110600 110600      ;TO BE SET IN RSDS FOR
2023 007604 104000 HLT          ;SEARCH FUNCTION SHOULD BE COMPLETED
2024 007606 022777 104230 171264 CMP          #104230,RSCSI ;SET RCSI
2025 007614 001401 BEQ          2$      ;SC IN RCSI SHOULD SET BECAUSE OF
2026 007616 104140 HLT          !DS!AS  ;COMPLETED SEARCH FUNCTION
2027 007620 016777 171336 171270 2$: MOV        UNITSV,RSAS ;CLEAR ATA
2028 007626 005777 171264 TST          RSAS    ;DID ATA CLEAR BY WRITING INTO IT?
2029 007632 001401 BEQ          3$      ;YES
2030 007634 104140 HLT          !DS!AS  ;RSAS SHOULD=0
2031 007636 022777 004230 171234 3$: CMP        #4230,RSCSI ;DID SC CLEAR BY CLEARING
2032 007644 001401 BEQ          +4      ;"ATA" YES
2033 007646 104140 HLT          !DS!AS  ;NO

```

```

2034 :*****
2035 :TEST 43          BLOCK SEARCH TEST 2
2036 :*****
2037 007650 104400 TST43: SCOPE
2038
2039 :MODULE TESTED: M7759, M7754, M7771, M7770
2040 :THIS TEST INITIALIZES A BLOCK SEARCH FUNCTION FOR SECTOR 0. WHEN THE DRIVE
2041 :IS CURRENTLY AT THE DESIRED SECTOR, THE BLOCK SEARCH FUNCTION
2042 :SHOULD NOT BE COMPLETED UNTIL THE DRIVE MAKES A COMPLETE REVOLUTION
2043 :AND REACHES THE BEGINNING OF THE DESIRED SECTOR.
2044
2045 007652 104414 MRSRC: CLDK          :CLEAR ALL REGISTERS
2046 007654 052767 000040 171304 BIS          #BITS,ONCEE :SET CLOCK FLAG
2047 007662 104416 PROMD          :PUT DRIVE INTO MAINTENANCE MOE
2048 007664 104420 MRCK          :CHECK MR REG
2049 007666 022701 22701          :TO EQUAL 22701
2050 007670 104424 MRINT          :INIT MR REG (CLEAR MRSP)
2051 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2052 007672 104430 MRIND          :
2053 007674 104420 MRCK          :CHECK MR REG TO EQUAL
2054 007676 022701 22701          :22701
2055 007700 104000 HLT          :
2056 ;STEP THRU RESYNC PERIOD
2057 007702 012767 001000 171272 MOV          #512,REPT
2058 007710 052767 000040 171250 BIS          #BITS,ONCEE
2059 007716 104422 MRRT1: MRCLK          :TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2060 007720 104420 MRCK          :CLOCK MR REG
2061 007722 072701 72701          :CHECK FOR
2062 007724 104000 HLT          :CORRECT DATA
2063 007726 104422 MRCLK          :MR = BAD GOOD = CORRECT DATA
2064 007730 104420 MRCK          :CLOCK MR REG
2065 007732 022701 22701          :CHECK FOR
2066 007734 104000 HLT          :CORRECT DATA
2067 007736 005367 171240 DEC          REPT
2068 007742 001365 BNE          MRRT1
2069 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE SP = 0
2070 007744 104422 MRCLK          :CLOCK MR REG
2071 007746 104420 MRCK          :MR SHOULD
2072 007750 072301 72301          :EQUALS 72301
2073 007752 104000 HLT          :MR=BAD GOOD=CORRECT ANS
2074 007754 104422 MRCLK          :CLOCK MR REG
2075 007756 104420 MRCK          :CHECK MR
2076 007760 022301 22301          :TO EQUAL 22301
2077 007762 104000 HLT          :MR=BAD GOOD=CORRECT ANS
2078 007764 012767 000100 171210 MOV          #100,REPT
2079 007772 104422 25: MRCLK          :STEP INTO SECTOR 0
2080 007774 005367 171202 DEC          REPT
2081 010000 001374 BNE          25
2082 010002 012777 000031 171070 45: MOV          #31,RSRCS1
2083 010010 104426 DSCK          :LOAD SEARCH COMMAND (M7759) FOR SECTOR 0
2084 010012 030400 30400          :CHECK RSDS
2085 010014 104000 HLT          :TO EQUAL 30400
2086 :PIP SHOULD BE SET AND DRY SHOULD
2087 010016 012767 021506 171156 MOV          #21506,REPT
:BE 0 FOR A DRIVE SEARCH CMD
:STEP 3 SECTORS BEYOND SECTOR 0

```



```

2088 010024 104422          1S:  MRCLK          ;CLOCK MR
2089 010026 104426          DSCK          ;RSDS SHOULD NOT
2090 010030 030400          30400        ;CHANGE TILL CLOCKING IS COMPLETED
2091 010032 104000          HLT          ;TO REACH SECTOR 3
2092 010034 005367 171142  DEC REPT      ;KEEP CLOCKING TILL
2093 010040 001371          BNE 1S       ;SECTOR 3 HAS BEEN REACHED
2094          ;ASSERT INDEX PULSE TO SIMULATE THE BEGINNING OF THE NEXT REVOLUTION
2095 010042 104430          MRIND
2096 010044 104420          MRCK          ;CHECK MR REG TO EQUAL
2097 010046 022701          22701
2098 010050 104000          HLT
2099
2100          ;STEP THRU RESYNC PERIOD
2101
2102 010052 012767 001000 171122  MOV #512.,REPT
2103 010060 052767 000040 171100  BIS #BITS,ONCEE
2104 010066 104422          MRWR1: MRCLK          ;TYPE OUT CLOCK COUNT IF AN ERROR OCCURS
2105 010070 104420          MRCK          ;CLOCK MR REG
2106 010072 072701          72701        ;CHECK FOR
2107 010074 104000          HLT          ;CORRECT DATA
2108 010076 104422          MRCLK          ;MR = BAD GOOD = CORRECT DATA
2109 010100 104420          MRCK          ;CLOCK MR REG
2110 010102 022701          22701        ;CHECK FOR
2111 010104 104000          HLT          ;CORRECT DATA
2112 010106 005367 171070  DEC REPT      ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2113 010112 001365          BNE MRWR1    ;FINISH LOOPING
2114          ;THROUGH RESYNC PERIOD
2115          ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2116          ;SP=0 EQUALS SECTOR PULSE
2117 010114 104422          MRCLK          ;CLOCK MR REG
2118 010116 104420          MRCK          ;MR SHOULD
2119 010120 072301          72301        ;EQUAL 72301
2120 010122 104000          HLT          ;MR=BAD GOOD=CORRECT ANS
2121 010124 104422          MRCLK          ;CLOCK MR REG
2122 010126 104420          MRCK          ;CHECK MR
2123 010130 022301          22301        ;TO EQUAL 22301
2124 010132 104000          HLT          ;MR=BAD GOOD=CORRECT ANS
2125
2126          ;NOTE ADD ONE MORE CLOCK PULSE TO LOOP COUNTER
2127 010134 104422          MRCLK          ;CLOCK MR REG
2128 010136 104426          DSCK          ;CHECK FOR "ATA" AND "DRY"
2129 010140 110600          110600      ;TO BE SET IN RSDS FOR
2130 010142 104000          HLT          ;SEARCH FUNCTION SHOULD BE COMPLETED
2131 010144 022777 104230 170726  CMP #104230,RS1
2132 010152 001401          BEQ 2S
2133 010154 104140          HLT !DS!AS
2134 010156 016777 171000 170732 2S:  MOV UNITSV,RSAS ;SC IN RSCS1 SHOULD SET BECAUSE OF
2135 010164 005777 170726          TST RSAS     ;COMPLETED SEARCH FUNCTION
2136 010170 001401          BEQ 3S
2137 010172 104140          HLT !DS!AS
2138 010174 022777 004230 170676 3S:  CMP #4230,RS1 ;CLEAR ATA
2139 010202 001401          BEQ +4       ;DID ATA CLEAR BY WRITING INTO IT?
2140 010204 104140          HLT !DS!AS ;NO ;YES
          ;RSAS SHOULD=0
          ;DID SC CLEAR BY CLEARING
          ;"ATA" YES

```

K04

```

2141 :*****
2142 :TEST 44 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSCS1)
2143 :*****
2144 010206 104400 TST44: SCOPE
2145
2146 :MODULE TESTED M7759, M7755, M7770
2147 :RMR ERROR IS CAUSED BY WRITTING INTO RSCS1 WHILE DOING A BLOCK SEARCH FUNCTION
2148 :CHECK RMR DECODER, E12, M7755, IF THIS TEST FAILS
2149
2150 010210 104414 RMRC1: CLDK ; CLEAR ALL DRIVE REGISTERS
2151 010212 042767 000040 170746 BIC #BITS,ONCEE ; CLEAR CLK CNT FLAG
2152 010220 104416 MRDM ; PUT DRIVE INTO MAINT MODE
2153 010222 104420 MRCK ; CHECK MR REG TO
2154 010224 022701 22701 ; EQUAL 22701
2155 010226 104424 MRINT ; INIT MAINT MODE (CLEAR MRSP)
2156 010230 012777 000001 170652 MOV #1,RSDA ; LOAD RSDA
2157 010236 012777 000031 170634 MOV #31,RSRCS1 ; LOAD BLOCK SEARCH FUNCTION
2158 010244 104426 DSCK ; CHECK RSDS
2159 010246 030400 30400 ; TO EQUAL 30400
2160 010250 104000 HLT ; DRY IN RSDS SHOULD BE
2161 ; CLEARED FOR DRIVE WAS
2162 ; ISSURED A BLOCK SEARCH FUNCTION
2163 ; RSDS=BAD GOOD=CORRECT ANS
2164 010252 012777 000011 170620 MOV #11,RSRCS1 ; LOAD A CLEAR FUNCTION
2165 ; THIS SHOULD CAUSE AN RMR
2166 ; ERROR FOR DRIVE WAS BUSY
2167 ; WHEN CLEAR COMMAND WAS GIVEN
2168 010260 012700 170630 MOV #RSER,BAD ; GET RSER REG
2169 010264 012701 000004 MOV #4,GOOD ; GET CORRECT ANS
2170 010270 020700 CMP GOOD,BAD ; DID RMR SET IN RSER?
2171 010272 001400 BEQ 1$ ; YES
2172 010274 104402 021435 TYPE ,TRMR ;
2173 010300 104402 010304 TYPE ,.+2 ; .ASCIZ "RSCS1"
2174 010312 104000 HLT ; RSER=BAD GOOD=CORRECT ANS
2175 010314 104426 1$: DSCK ; CHECK RSDS TO
2176 010316 150600 150600 ; EQUAL 150600
2177 010320 104000 HLT ; RSDS=BAD GOOD=CORRECT ANS
2178 010322 022777 104230 170550 CMP #104230,RSRCS1 ; DID CORRECT BITS SET IN RSCS1
2179 010330 001401 BEQ 2$ ; YES
2180 010332 104040 HLT !DS ; RSCS1 SHOULD=104230
2181 ; RSDS SHOULD=150600
2182 ; RSER SHOULD=4
2183 010334 022777 000001 170546 2$: CMP #1,RSDA ; DID CLR CLEAR RSDA
2184 010342 001401 BEQ 4$ ; NO
2185 010344 104004 HLT !DA ; RSDA SHOULD=1
2186 010346 104414 4$: CLDK ; CLEAR ALL REGISTERS
2187 010350 005777 170540 TST #RSER ; RSER SHOULD CLEAR
2188 010354 001401 BEQ 3$ ; RSER OK
2189 010356 104040 HLT !DS ; RSER SHOULD=0 FOR THE
2190 ; CLEAR BIT WAS LOADED IN RSCS2
2191 010360 022777 004200 170512 3$: CMP #4200,RSRCS1 ; RSCS1 SHOULD=4200 FOR THE
2192 010366 001401 BEQ .+4 ; CLEAR BIT WAS LOADED IN RSCS2
2193 010370 104040 HLT !DS ; RSCS1 SHOULD=4200

```

L04

MAINDEC-11-DERSD-C RH70-RSD4 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 51
 DERSDC.SRC 17-MAY-77 14:16 TST45 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)

```

2194 ;*****
2195 ;TEST 45 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSDA)
2196 ;*****
2197 010372 104400 TST45: SCOPE
2198
2199 ;MODULE TESTED M7755 M7759 M7770
2200 ;RMR ERROR IS CAUSED BY WRITTING INTO RSDA WHILE DOING A BLOCK SEARCH FUNCTION
2201
2202 010374 104414 RMRC2: CLRDK ;CLEAR ALL DRIVE REGISTERS
2203 010376 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2204 010400 104420 MRCK ;CHECK MR REG TO
2205 010402 022701 22701 ;EQUAL 22701
2206 010404 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2207 010406 012777 000001 170474 MOV #1,RSDA ;LOAD RSDA
2208 010414 012777 000031 170456 MOV #31,RSRCS1 ;LOAD BLOCK SEARCH FUNCTION
2209 010422 104426 DSKC ;CHECK RSDS
2210 010424 030400 30400 ;TO EQUAL 30400
2211 010426 104000 HLT ;DRY IN RSDS SHOULD BE
2212 ;CLEARED FOR DRIVE WAS
2213 ;ISSURED A BLOCK SEARCH FUNCTION
2214 ;RSDS=BAD GOOD=CORRECT ANS
2215 010430 005077 170454 CLR RSDA ;MODIFY RSDA
2216 ;THIS SHOULD CAUSE AN RMR
2217 ;ERROR FOR DRIVE WAS BUSY
2218 ;WHEN COMMAND WAS GIVEN
2219 010434 017700 170454 MOV RSR, BAD ;GET RSR REG
2220 010440 012701 000004 MOV #4, GOOD ;GET CORRECT ANS
2221 010444 020100 CMP GOOD, BAD ;DID RMR SET IN RSR?
2222 010446 001410 BEQ IS ;YES
2223 010450 104402 021435 TYPE , TRMR ;.ASCIZ "RSDA"
2224 010454 104402 010460 TYPE ,.+2 ;RSDA=BAD GOOD=CORRECT ANS
2225 010466 104000 HLT ;CHECK RSDS TO
2226 010470 104426 1S: DSKC ;EQUAL 150600
2227 010472 150600 150600 HLT ;RSDS=BAD GOOD=CORRECT ANS
2228 010474 104000 HLT ;DID CORRECT BITS SET IN RSCS1
2229 010476 022777 104230 170374 CMP #104230,RSRCS1 ;YES
2230 010504 001401 BEQ 2S ;RSCS1 SHOULD=104230
2231 010506 104040 HLT !DS ;RSDS SHOULD=50400
2232 ;RSER SHOULD=4
2233 ;DID CLR CLEAR RSDA
2234 010510 022777 000001 170372 2S: CMP #1,RSDA ;NO
2235 010516 001401 BEQ 4S ;RSDA SHOULD=1
2236 010520 104004 HLT !DA ;CLEAR ALL REGISTERS
2237 010522 104414 4S: CLRDK ;RSDA SHOULD CLEAR
2238 010524 005777 170364 TST RSR ;RSER SHOULD CLEAR
2239 010530 001401 BEQ 3S ;RSER OK
2240 010532 104040 HLT !DS ;RSER SHOULD=0 FOR THE
2241 ;CLEAR BIT WAS LOADED IN RSCS2
2242 010534 022777 004200 170336 3S: CMP #4200,RSRCS1 ;RSCS1 SHOULD=4200 FOR THE
2243 010542 001401 BEQ +4 ;CLEAR BIT WAS LOADED IN RSCS2
2244 010544 104040 HLT !DS ;RSCS1 SHOULD=4200
2245
  
```

M04

MAINDEC-11-DERSO-C
DERSOC.SRC

17-MAY-77 14:16

RH70-RS04 MAINTENANCE MODE DIAGNOSTIC

MACY11 27(1006) 17-MAY-77 15:36 PAGE 52
TST46 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)

```

2246 *****
2247 :TEST 46 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSER)
2248 *****
2249 TST46: SCOPE
2250
2251 :MODULE TESTED M7759, M7755, M7770
2252 :RMR ERROR IS CAUSED BY WRITTING INTO RSER WHILE DOING A BLOCK SEARCH FUNCTION
2253 :CHECK RMR DECODER, E12-M7755, IF THIS TEST FAILS.
2254
2255 RMRC3: CLDK          ;CLEAR ALL DRIVE REGISTERS
2256 010550 104414      BIC          #BITS,ONCEE ;CLEAR CLOCK COUNT FLAG
2257 010552 042767 000040 170406 MROMD        ;PUT DRIVE INTO MAINT MODE
2258 010560 104416      MRCK          ;CHECK MR REG TO
2259 010562 104420      MRCK          ;CHECK MR REG TO
2260 010564 022701      22701        ;EQUAL 22701
2261 010566 104424      MRINT        ;INIT MAINT MODE (CLEAR MRSP)
2262 010570 012777 000001 170312 MOV          #1,RSOA ;LOAD RSOA
2263 010576 012777 000031 170274 MOV          #31,RSOS1 ;LOAD BLOCK SEARCH FUNCTION
2264 010604 104426      DSCK          ;CHECK RSDS
2265 010606 030400      30400        ;TO EQUAL 30400
2266 010610 104000      HLT          ;DRY IN RSDS SHOULD BE
2267                                     ;CLEARED FOR DRIVE WAS
2268                                     ;ISSUED A BLOCK SEARCH FUNCTION
2269 010612 012777 177777 170274 MOV          #-1,RSER ;RSDS=BAD GOOD=CORRECT ANS
2270                                     ;MODIFY RSER
2271                                     ;THIS SHOULD CAUSE AN RMR
2272                                     ;ERROR FOR DRIVE WAS BUSY
2273                                     ;WHEN COMMAND WAS GIVEN
2273 010620 017700 170270      MOV          RSER,BAD ;GET RSER REG
2274 010624 012701 000004      MOV          #4,GOOD ;GET CORRECT ANS
2275 010630 020100      CMP          GOOD,BAD ;DID RMR SET IN RSER?
2276 010632 001410      BEQ          IS ;YES
2277 010634 104402 021435      TYPE          ,TRMR ;ASCIZ "RSER"
2278 010640 104402 010644      TYPE          ,.+2 ;RSER=BAD GOOD=CORRECT ANS
2279 010652 104000      HLT          ;CHECK RSDS TO
2280 010654 104426      IS:        DSCK          ;EQUAL 150600
2281 010656 150600      150600      ;RSDS=BAD GOOD=CORRECT ANS
2282 010660 104000      HLT          ;DID CORRECT BITS SET IN RSCS1
2283 010662 022777 104230 170210 CMP          #104230,RSOS1 ;YES
2284 010670 001401      BEQ          4S ;RSCS1 SHOULD=104230
2285 010672 104040      HLT          !DS ;RSDS SHOULD=150600
2286                                     ;RSER SHOULD=4
2287                                     ;CLEAR ALL REGISTERS
2288 010674 104414      4S:        CLDK          ;RSER SHOULD CLEAR
2289 010676 005777 170212      TST          RSER ;RSER OK
2290 010702 001401      BEQ          3S ;RSER SHOULD=0 FOR THE
2291 010704 104040      HLT          !DS ;CLEAR BIT WAS LOADED IN RSCS2
2292                                     ;RSCS1 SHOULD=4200 FOR THE
2293 010706 022777 004200 170164 3S: CMP          #4200,RSOS1 ;CLEAR BIT WAS LOADED IN RSCS2
2294 010714 001401      BEQ          +4 ;RSCS1 SHOULD=4200
2295 010716 104040      HLT          !DS

```

N04

```

2296
2297
2298
2299 010720 104400
2300
2301
2302
2303
2304
2305 010722 104414
2306 010724 104416
2307 010726 104420
2308 010730 022701
2309 010732 104424
2310 010734 012777 000001 170146
2311 010742 012777 000031 170130
2312 010750 104426
2313 010752 030400
2314 010754 104000
2315
2316
2317
2318 010756 005077 170134
2319
2320
2321 010762 017700 170126
2322 010766 012701 000000
2323 010772 020100
2324 010774 001435
2325 010776 104402 011002
2326 011066 104000
2327 011070 104426
2328 011072 030400
2329 011074 104000
2330 011076 022777 004231 167774
2331 011104 001401
2332 011106 104040
2333
2334
2335 011110 104414
2336 011112 005777 167776
2337 011116 001401
2338 011120 104040
2339
2340 011122 022777 004200 167750
2341 011130 001401
2342 011132 104040

```

```

:*****
:TEST 47 DISK REGISTER MODIFIED REFUSED (RMR) ERROR TEST (RSAS)
:*****
TST47: SCOPE

:MODULE TESTED: M7759, M7755, M7770
:RMR ERROR SHOULD NOT SET BY WRITTING INTO RSAS WHILE DOING A BLOCK SEARCH FUNCTION
:IF TEST FAILS, CHECK RMR DECODER E12-M7755.

RMR4: CLRDK MRDMD MRCK 22701 MRINT MOV #1,RSDA MOV #31,RSCS1 DSCK 30400 HLT

: CLEAR ALL DRIVE REGISTERS
: PUT DRIVE INTO MAINT MODE
: CHECK MR REG TO
: EQUAL 22701
: INIT MAINT MODE (CLEAR MRSP)
: LOAD RSDA
: LOAD BLOCK SEARCH FUNCTION
: CHECK RSDS
: TO EQUAL 30400
: DRY IN RSDS SHOULD BE
: CLEARED FOR DRIVE WAS
: ISSURED A BLOCK SEARCH FUNCTION
: RSDS=BAD GOOD=CORRECT ANS
: WRITE INTO ATTENTION SUMMARY REGISTER.
: SHOULD BE NO RMR ERROR BECAUSE
: WRITING RSAS IS ALLOWED ANYTIME.

MOV #RSER,BAD
MOV #0,GOOD
CMP GOOD,BAD
BEQ 1$
TYPE ,.+2
1$: DSCK 30400 HLT
CMP #4231,RSCS1
BEQ 4$
HLT !DS

4$: CLRDK TST #RSER BEQ 3$ HLT !DS

3$: CMP #4200,RSCS1 BEQ .+4 HLT !DS

```

```

2343 :*****
2344 :TEST 50 DRIVE SELECT TEST
2345 :*****
2346 011134 104400 TST50: SCOPE
2347
2348 :MODULE TESTED: M7755
2349 :THE PROGRAM LOADS A DRIVE REGISTER, OF THE DRIVE UNDER TEST, TO ALL ONES.
2350 :THE PROGRAM THEN FINDS A NON-EXISTENT DRIVE AND TRIES TO LOAD ITS
2351 :REGISTER WITH ALL ZEROS. THIS SHOULD CAUSE "NED" TO
2352 :SET IN RSCS2. THE PROGRAM RE-SELECTS THE DRIVE UNDER TEST AND CHECKS
2353 :ITS REGISTER TO SEE IF IT WAS MODIFIED. IT SHOULD CONTAIN ALL ONES.
2354 :CHECK UNIT NO. COMPARATOR, E19-M7755 IF TEST FAILS
2355
2356 011136 104414 MRDSEL: CLRDK ;CLEAR ALL REGISTERS
2357 011140 104416 MRDMD ;PUT DRIVE INTO MAINT MODE
2358 011142 104420 MRCK ;CHECK MAINT REG
2359 011144 022701 22701 ;TO EQUAL 22701
2360 011146 104424 MRINT ;INITIALIZE MAINT MODE (CLEAR MRSP)
2361 ;BY SENDING 2 CLOCK PULSES
2362 011150 012777 177777 167732 ;LOAD DISK ADDR REG OF DRIVE UNDER TEST
2363
2364 011156 012767 000401 170030 ;SEARCH FOR NON EXISTENT DRIVES
2365 011164 005001 ;
2366 011166 010177 167710 1S: MOV #401,WORK
2367 011172 005777 167716 ;GOOD
2368 011176 032777 010000 167676 ;GOOD, RSCS2
2369 011204 001005 ;RSCS2
2370 011206 005201 ;BIT12, RSCS2
2371 011210 006167 170000 ;ZS
2372 011214 103460 ;BNE ZS
2373 011216 000763 ;INC GOOD
2374 011220 012777 004000 167652 2S: ROL WORK
2375 011226 010167 167766 ;BCS NEDDON
2376 011232 010177 167644 ;BR 1S
2377 011236 005077 167646 ;MOV #4000, RSCS1
2378 ;MOV GOOD, WORK1
2379 ;MOV GOOD, RSCS2
2380 011242 017700 167634 ;CLR RSDA
2381 011246 052701 010100 ;MOV RSCS2, BAD
2382 ;BIS #10100, GOOD
2383 011252 020100 ;CMP GOOD, BAD
2384 011254 001401 ;BEQ .+4
2385 011256 104000 ;HLT
2386 ;RSCS2=BAD GOOD=CORRECT ANS
2387 011260 022777 160200 167612 ;CMP #160200, RSCS1
2388 011266 001401 ;BEQ .+4
2389 011270 104004 ;HLT ;DA
2390
2391

```

2392	011272	005777	167620		TST	RSCAS			:DID ANY ATTENTION BITS SET?
2393	011276	001401			BEQ	.+4			:NO
2394	011300	104100			HLT	!AS			:NO ATTENTION BITS SHOULD BE SET
2395	011302	112777	000100	167624	MOV8	#100,RSCS18			:CLEAR TRE
2396	011310	032777	010000	167564	BIT	#NED,RSCS2			:DID NED CLEAR
2397	011316	001401			BEQ	.+4			:YES
2398	011320	104040			HLT	!DS			:NED DID NOT CLEAR IN RSCS2
2399									:BY CLEARING TRE BIT IN RSCS1
2400	011322	016777	167632	167552	MOV	UNNUM,RSCS2			:LOAD CORRECT UNIT NUMBER
2401	011330	022777	177777	167552	CMF	#-1,RSDA			:DID RSDA GET MODIFIED
2402									:WHILE WRITING INTO A NON
2403									:EXISTENT DRIVE?
2404	011336	001443			BEQ	NNDD			:NO
2405	011340	104004			HLT	!DA			:RSDA SHOULD= -1
2406	011342	016700	167652		MOV	WORK1,BAD			:IT GOT MODIFIED WHILE WRITING
2407	011346	016701	167606		MOV	UNNUM,GOOD			:INTO A NED
2408	011352	104000			HLT				:GOOD=DRIVE UNDER TEST
2409	011354	000434			BR	NNDD			:BAD=NON EXISTENT DRIVE THAT WAS
2410									:IN RSCS2 WHEN RSDA GOT MODIFIED
2411	011356	032767	010000	167602	NEDDON:	BIT	#BIT12,ONCEE		:WAS THIS TYPED BEFORE?
2412	011364	001030				BNE	NNDD		:YES
2413	011366	104402	011372			TYPE	.+2		:ASCIZ <15><12>"COULD NOT FIND A NON-EXISTENT DRIVE"
2414	011440	052767	010000	167520		BIS	#BIT12,ONCEE		:SET TYPED MESSAGE FLAG
2415	011446				NNDD:				

24616
24617
24618
24619
24620
24621
24622
24623
24624
24625
24626
24627
24628
24629
24630
24631
24632
24633
24634
24635
24636
24637
24638
24639
24640
24641
24642
24643
24644
24645
24646
24647
24648
24649
24650
24651
24652
24653
24654
24655
24656
24657
24658
24659
24660
24661
24662
24663
24664
24665
24666
24667
24668
24669

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 026610
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 026610 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  
CLRK ;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, #RSBA ;LOAD BUS ADDR REG  
MOV #177600, #RSWC ;LOAD WORD COUNT REG  
MOV #61, #RSCS1 ;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
```


E05

```

2470                                   ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2471 011600 104430                   MRIND
2472 011602 104420                   MRCK                               ;CHECK MR REG TO EQUAL
2473 011604 020501                   20501                           ;20501 FOR A
2474 011606 104000                   HLT                               ;WRITE COMD HAS BEEN ISSUED
2475
2476                                   ;STEP THRU RESYNC PERIOD
2477
2478 011610 012767 001000 167364      MOV           #512,REPT
2479 011616 052767 000040 167342      BIS           #BITS,ONCEE       ;TYPE OUT CLOCK COUNT IF ERROR OCCURS
2480 011624 104422                   MRWRT1: MRCLK                   ;CLOCK MR REG
2481 011626 104420                   MRCK                           ;CHECK FOR
2482 011630 070501                   70501                           ;CORRECT DATA
2483 011632 104000                   HLT                           ;MR = BAD GOOD = CORRECT DATA
2484 011634 104422                   MRCLK                       ;CLOCK MR REG
2485 011636 104420                   MRCK                           ;CHECK FOR
2486 011640 020501                   20501                           ;CORRECT DATA
2487 011642 104000                   HLT                           ;ERROR WHILE CLOCKING THROUGH RESYNC PERIOD
2488 011644 005367 167332           DEC           REPT               ;FINISH LOOPING
2489 011650 001365                   BNE           MRWRT1           ;THROUGH RESYNC PERIOD
2490
2491                                   ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2492                                   ;SP=0 EQUALS SECTOR PULSE
2493 011652 104422                   MRCLK                       ;CLOCK MR REG
2494 011654 104420                   MRCK                           ;MR SHOULD
2495 011656 070101                   70101                           ;EQUAL 70101
2496 011660 104000                   HLT                           ;MR=BAD GOOD=CORRECT ANS
2497 011662 104422                   MRCLK                       ;CLOCK MR REG
2498 011664 104420                   MRCK                           ;CHECK MR
2499 011666 020101                   20101                           ;TO EQUAL 20101
2500 011670 104000                   HLT                           ;MR=BAD GOOD=CORRECT ANS
2501
2502                                   ;PERFORM 63 DOUBLE MAINT CLOCK OPERATIONS--WRITING PREAMBLE
2503
2504 011672 012767 000077 167302      MOV           #63.,REPT
2505 011700 104422                   MRWRT2: MRCLK                   ;CLOCK MR REG
2506 011702 104420                   MRCK                           ;CHECK MR REG
2507 011704 071501                   71501                           ;TO EQUAL 71501
2508 011706 104000                   HLT                           ;MR=BAD GOOD=CORRECT ANS
2509 011710 104422                   MRCLK                       ;CLOCK MR REG
2510 011712 104420                   MRCK                           ;CHECK MR REG
2511 011714 021501                   21501                           ;TO EQUAL 21501
2512 011716 104000                   HLT                           ;MR=BAD GOOD=CORRECT ANS
2513 011720 005367 167256           DEC           REPT               ;DONE YET
2514 011724 001365                   BNE           MRWRT2           ;NO LOOP
  
```

F05

;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN

2515									
2516									
2517	011726	104422			MRCLK			:CLOCK MR REG	
2518	011730	104420			MRCK			:CHECK MR REG	
2519	011732	171501			171501			:TO EQUAL 171501	
2520	011734	104000			HLT			:MR REG=BAD GOOD=CORRECT ANS	
2521	011736	104422			MRCLK			:CLOCK MR REG	
2522	011740	104420			MRCK			:MR REG SHOULD	
2523	011742	025501			25501			:EQUAL 25501	
2524	011744	104000			HLT			:MR REG=BAD GOOD=CORRECT ANS	
2525	011746	104422			MRCLK				
2526	011750	104420			MRCK				
2527	011752	175501			175501				
2528	011754	104000			HLT				
2529									
2530	011756	012767	000003	167216					
2531	011764	104422			MRWRT3: MRCLK			:CLOCK MR REG	
2532	011766	104420			MRCK			:CHECK MR REG	
2533	011770	025501			25501			:TO EQUAL 25501	
2534	011772	104000			HLT			:MR=BAD GOOD=CORRECT ANS	
2535	011774	104422			MRCLK			:CLOCK MR REG	
2536	011776	104420			MRCK			:CHECK MR REG	
2537	012000	175501			175501			:TO EQUAL 175501	
2538	012002	104000			HLT			:MR REG=BAD GOOD=CORRECT ANS	
2539	012004	005367	167172		DEC	REPT		:DONE YES?	
2540	012010	001365			BNE	MRWRT3		:NO LOOP BACK	
2541									
2542									
2543									
2544	012012	104422			MRCLK			:CLOCK MR REG	
2545	012014	104420			MRCK			:CHECK MR REG	
2546	012016	027501			27501			:TO EQUAL 27501	
2547	012020	104000			HLT			:MR=BAD GOOD=CORRECT ANS	
2548	012022	104422			MRCLK			:CLOCK MR REG	
2549	012024	104420			MRCK			:MR REG SHOULD	
2550	012026	123501			123501			:EQUAL 123501	
2551	012030	104000			HLT			:MR=BAD GOOD=CORRECT ANS	
2552									
2553									
2554									
2555	012032	104422			MRCLK			:CLOCK MR REG	
2556	012034	104420			MRCK			:MR REG SHOULD NOW	
2557	012036	073501			73501			:EQUAL 73501	
2558	012040	104000			HLT			:MR=BAD GOOD=CORRECT ANS	
2559	012042	012705	026610		MOV	#INBUF,R5		:GET STARTING ADDR FOR DATA BUFFER	
2560	012046	011504			MOV	(R5),R4		:GET DATA	

;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE

;MOVE DATA WORD INTO R504 SHIFT REGISTER (M7753)

;ENCODE SYNC 1 (M7751)

G05

MAINDEC-11-DERSD-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 59
 DERSDC.SRC 17-MAY-77 14:16 TST51 MAINTENANCE MODE WRITE TEST

2561	012050	012767	002167	167136		MOV	#1143.,WORK	:DOING A 1 SECTOR TRANSFER 127 WORDS
2562								:18 BITS PER WORD-CLOCK LOOPS
2563								:TAKE CARE OF 2 BITS AT A TIME
2564								:127 TIMES 9 EQUALS 1143 LOOPS
2565								:TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
2566	012056	042767	000200	167102		BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
2567	012064	052767	000100	167074		BIS	#BIT6,ONCEE	:SET 1ST TRANSFER WORD FLAG
2568	012072	104432			1S:	X9IT		:GET 2 BITS OF DATA
2569	012074	104434				CLKD1		:SEND FIRST CLOCK PULSE
2570								:AND CALCULATE MR REG
2571								:FOR CORRECT DATA (MMDT+MMD8)
2572	012076	104000				HLT		:MR REG NOT CORRECT
2573	012100	104436				CLKD2		:SEND 2ND CLOCK PULSE TO
2574								:COMPLETE TRANSFER OF 2 BITS
2575								:CALCULATE CORRECT ANS FOR
2576								:MR REG (MMDT+MMD8)
2577	012102	104000				HLT		:MR=BAD GOOD=CORRECT ANS
2578	012104	032767	000200	167054		BIT	#BIT7,ONCEE	:ON LAST WORD YET?
2579	012112	001015				BNE	2S	:YES
2580	012114	032767	000400	167044		BIT	#BIT8,ONCEE	:ON CRC WORD YET?
2581	012122	001040				BNE	3S	:YES
2582	012124	005367	167064			DEC	WORK	:DONE WITH 127 WORDS?
2583	012130	001360				BNE	1S	:NO
2584								
2585	012132	052767	000200	167026		BIS	#BIT7,ONCEE	:SET LAST WORD FLAG
2586	012140	012767	000012	167046		MOV	#10.,WORK	:SET UP TO TRANSFER LAST WORD
2587	012146	005367	167042		2S:	DEC	WORK	:DONE YET?
2588	012152	001347				BNE	1S	:NO
2589								
2590	012154	052767	000400	167004		BIS	#BIT8,ONCEE	:SET TRANSFERRING CRC WORD
2591	012162	042767	000200	166776		BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
2592	012170	004767	011262			JSR	PC,GENCRC	:GENERATE CRC WORD
2593								:AND LEAVE IN "WORK"
2594	012174	012702	026610			MOV	#INBUF,R2	:GO TO END
2595	012200	062702	000400			ADD	#400,R2	:OF DATA BUFFER
2596	012204	016712	167004			MOV	WORK,R2	:LOAD CRC WORD
2597	012210	010205				MOV	R2,R5	:RESET POINTER FOR
2598	012212	162705	000002			SUB	#2,R5	:R5 FOR CRC WD
2599	012216	012767	000012	166770		MOV	#10.,WORK	:SETUP TO XFER CRC
2600	012224	005367	166764		3S:	DEC	WORK	:DONE YET
2601	012230	001320				BNE	1S	:NO

H05

MAINDEC-11-DERSO-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 60
 DERSDC.SRC 17-MAY-77 14:16 TST51 MAINTENANCE MODE WRITE TEST

```

2602                ;EBL SHOULD NOW ASSERT
2603
2604 012232 104422          MRCLK                ;CLOCK MR REG TO STOP THROUGH
2605                                     ;THE RS04 DISK SECTOR DEAD BAND AREA
2606 012234 104420          MRCK                ;CHECK MR REG
2607 012236 153501          153501             ;TO EQUAL 103501
2608 012240 104000          HLT                ;MR REG=BAD GOOD=CORRECT ANS
2609
2610                ;LOOP 6 TIMES
2611
2612 012242 012767 000006 166732 4S:  MOV      #6,REPT
2613 012250 104422          MRCLK                ;CLOCK MR REG
2614 012252 104420          MRCK                ;CHECK MR REG
2615 012254 003501          3501             ;TO EQUAL 53501
2616 012256 104000          HLT                ;MR=BAD GOOD=CORRECT ANS
2617 012260 104422          MRCLK                ;CLOCK MR REG
2618 012262 104420          MRCK                ;CHECK MR REG
2619 012264 153501          153501             ;TO EQUAL 153501
2620 012266 104000          HLT                ;MR=BAD GOOD=CORRECT ANS
2621 012270 005367 166706  DEC      REPT
2622 012274 001365          BNE      4S        ;DONE LOOPING YET?
2623                                     ;NO
2624
2625                ;FINISH UP
2626
2626 012276 104422          MRCLK                ;CLOCK MR REG
2627 012300 104420          MRCK                ;CHECK MR REG
2628 012302 003501          3501             ;TO EQUAL 3501
2629 012304 104000          HLT                ;MR REG=BAD GOOD=CORRECT ANS
2630 012306 104422          MRCLK                ;CLOCK MR REG
2631 012310 104420          MRCK                ;CHECK MR REG
2632 012312 151501          151501             ;TO EQUAL 151501
2633 012314 104000          HLT                ;MR=BAD GOOD=CORRECT ANS
2634
2635                ;TRANSFER SHOULD NOW BE COMPLETE
2636
2637 012316 104422          MRCLK                ;CLOCK MR REG
2638 012320 104420          MRCK                ;CHECK MRE REG
2639 012322 002701          2701             ;TO EQUAL 2701
2640 012324 104000          HLT                ;MR=BAD GOOD=CORRECT ANS
2641
2642                ;NOW TEST CONTROLLER
2643
2644 012326 005777 166546  TST      @RSCS1    ;ANY ERRORS?
2645 012332 100001          BPL      5S        ;NO
2646 012334 104014          HLT      !DA!WC   ;YES
2647 012336 005777 166542 5S:  TST      @RSWC    ;DID WC GO TO 0
2648 012342 001401          BEQ     +4        ;YES
2649 012344 104010          HLT     !WC       ;WC SHOULD BE = TO 0
2650 012346 022777 000001 166534 CMP     #1,@RSDA  ;DOES RSDA=1
2651 012354 001401          BEQ     +4        ;YES
2652 012356 104004          HLT     !DA       ;RSDA SHOULD=1
2653 012360 032767 000002 166556 BIT     #BIT1,FLAG2 ;IN MAINT VERIFY TEST
2654 012366 001002          BNE     +6        ;NO
2655 012370 000137 020720  JMP     @#MRVR2   ;YES, GO TO VERIFY TEST
  
```

```

2656 :*****
2657 :TEST 52 MAINTENANCE READ TEST
2658 :*****
2659 012374 104400 TST52: SCOPE
2660
2661 :MODULE TESTED: M7771, M7753, M7751
2662 :THIS IS AN RS04 DISK MAINTENANCE MODE (SINGLE-STEPPED) SECTOR
2663 :READ TEST. WE ARE TESTING THE COMPLETE DATA PATH FROM THE
2664 :DISK DECODING LOGIC TO CORE MEMORY. (THE PHASE LOCK LOOP IS
2665 :NOT TESTED IN MAINTENANCE MODE.)
2666
2667 012376 104414 MRD: CLDK ;CLEAR DRIVE REGISTERS
2668 012400 052767 000040 166560 BIS #BITS ONCEE ;SET TYPE CLOCK COUNT FLAG
2669 012406 042767 147716 166552 BIC #147716,ONCEE ;CLEAR ALL OTHER FLAG BITS
2670 012414 104430 MRIND ;SEND INDEX PULSE TO MR REG
2671 012416 104420 MRCK ;CHECK MR REG
2672 012420 022701 22701 ;TO EQUAL 22701
2673 012422 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2674 ;BY SENDING 2 CLOCK PULSES
2675
2676 012424 005067 166514 CLR FLAG2 ;CLEAR FLAG TEST BITS
2677
2678 :FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
2679 :DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
2680 : A WORD OF ALL 1'S
2681 : FLOATING 1'S PATTERN (16 WORDS)
2682 : A PATTERN OF 146314 (110 WORDS)
2683
2684 012430 012702 026610 MOV #INBUF,R2 ;GET LOCATION OF INBUF
2685 012434 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
2686 012436 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
2687 012442 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
2688 012444 000261 SEC ;A PATTERN OF FLOATING ONES
2689 012446 006103 15: ROL R3 ;GET PATTERN
2690 012450 103402 BCS 25 ;DONE GET OUT
2691 012452 010322 MOV R3,(R2)+ ;FILL BUFFER
2692 012454 000774 BR 15 ;CONT
2693 012456 012703 000156 25: MOV #110,R3 ;FILL REMAINING PORTION OF
2694 012462 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
2695 012466 010422 35: MOV R4,(R2)+ ;LOAD BUFFER
2696 012470 005303 DEC R3 ;DONE YET
2697 012472 001375 BNE 35 ;NO
2698
2699 :NOTE:
2700 :INBUF CONTAINS THE TABLE OF DATA WHICH IS "READ".
2701 :VIA THE MRDB AND MRDT BITS IN RSMR
2702 :OUTBUF IS WHERE THE DATA WORDS FROM THE
:MASSBUS ARE STORED

```

J05

MAINDEC-11-DERSD-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 62
 DERSDC.SRC 17-MAY-77 14:16 TST52 MAINTENANCE READ TEST

```

2703                                   ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2704
2705   012474   012777   027410   166404           MOV       #OUTBUF,ARSBA       ;LOAD BUSS ADDR REG
2706   012502   012777   177600   166374           MOV       #177600,ARSMC       ;LOAD WORD COUNT REG
2707   012510   012777   000071   166362           MOV       #71,ARSCS1       ;LOAD READ COMMAND
2708   012516   012702   000200           MOV       #200,R2           ;CLEAR THE OUTBUF TABLE
2709   012522   012703   027410           MOV       #OUTBUF,R3       ;SO THAT WHEN THE READ
2710   012526   005023                                   4S:   CLR       (R3)+           ;IS FINISHED, WE CAN COMPARE
2711   012530   005302                                   DEC       R2               ;WHAT WE GOT (OUTBUF) WITH WHAT
2712   012532   001375                                   BNE       4S               ;WE EXPECTED (INBUF).
2713   012534   104446                                   GETSP                   ;CLOCK ROUTINE TO GET SECTOR PULSE
2714                                                                                                           ;TO CLEAR OUT COUNTERS AND REGISTERS
2715                                                                                                           ;THAT OTHERWISE COULD NOT BE CLEARED.
2716   012536   104220                                   HLT       !MR             ;CLOCK MR REG 2 TIMES SP = 1
2717   012540   104450                                   SPASS
2718
2719                                   ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2720   012542   104430                                   MRIND
2721   012544   104420                                   MRCK                     ;CHECK MR REG TO EQUAL
2722   012546   022601                                   22601                   ;22601 FOR A
2723   012550   104000                                   HLT                     ;READ COMD
2724
2725                                   ;STEP THRU RESYNC PERIOD
2726
2727   012552   012767   001000   166422           MOV       #512,REPT
2728   012560   052767   000040   166400           BIS       #BITS,ONCEE     ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
2729   012566   104422                                   MRRD1:  MRCLK             ;CLOCK MR REG
2730   012570   104420                                   MRCK                   ;CHECK FOR
2731   012572   072601                                   72601                   ;CORRECT DATA
2732   012574   104000                                   HLT                     ;MR=BAD GOOD=CORRECT DATA
2733   012576   104422                                   MRCLK                   ;CLOCK MR REG
2734   012600   104420                                   MRCK                   ;CHECK FOR
2735   012602   022601                                   22601                   ;CORRECT DATA
2736   012604   104000                                   HLT                     ;ERROR WHILE CLOCKING THROUGH RESYNC
2737   012606   005367   166370           DEC       REPT
2738   012612   001365                                   BNE       MRRD1         ;FINISH LOOPING
2739                                                                                                           ;THROUGH RESYNC PERIOD
2740
2741                                   ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2742                                   ;SP=0 EQUALS SECTOR PULSE
2742   012614   104422                                   MRCLK                   ;CLOCK MR REG
2743   012616   104420                                   MRCK                   ;MR SHOULD
2744   012620   072201                                   72201                   ;EQUAL 72201
2745   012622   104000                                   HLT                     ;MR=BAD GOOD=CORRECT ANS
2746   012624   104422                                   MRCLK                   ;CLOCK MR REG
2747   012626   104420                                   MRCK                   ;CHECK MR
2748   012630   022201                                   22201                   ;TO EQUAL 22201
2749   012632   104000                                   HLT                     ;MR=BAD GOOD=CORRECT ANS
  
```

K05

```

2750                                   ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
2751
2752 012634 012767 000107 166340   MRRD2:  MOV       #71.,REPT                   ;CLOCK MR REG
2753 012642 104422                   MRCK                   ;CHECK MR REG
2754 012644 104420                   73601                   ;TO EQUAL 73601
2755 012646 073601                   HLT                    ;MR=BAD GOOD=CORRECT ANS
2756 012650 104000                   MRCK                   ;CLOCK MR REG
2757 012652 104422                   MRCK                   ;CHECK MR REG
2758 012654 104420                   23601                   ;TO EQUAL 23601
2759 012656 023601                   HLT                    ;MR=BAD GOOD=CORRECT ANS
2760 012660 104000                   DEC       REPT           ;DONE YET
2761 012662 005367 166314           BNE       MRRD2           ;NO LOOP
2762 012666 001365                   MRCK                   ;CLOCK MR REG
2763 012670 104422                   MRCK                   ;CHECK MR REG
2764 012672 104420                   73601                   ;TO EQUAL 73601
2765 012674 073601                   HLT                    ;MR=BAD GOOD=CORRECT ANS
2766 012676 104000
2767
2768                                   ;READ SYNC"1"
2769
2770 012700 012777 000055 166216       MOV       #55,RSMR
2771 012706 012777 000045 166210       MOV       #45,RSMR
2772 012714 104420                   MRCK                   ;CHECK MR REG
2773 012716 023645                   23645                   ;TO EQUAL
2774 012720 104000                   HLT                    ;CONTENTS OF GOOD
2775 012722 012777 000055 166174       MOV       #55,RSMR
2776 012730 012777 000045 166166       MOV       #45,RSMR
2777 012736 104420                   MRCK
2778 012740 173645                   173645
2779 012742 104000                   HLT
2780
2781                                   ;READ DATA
2782 012744 005067 166254           MRRD3:  CLR       WORK3                   ;CLEAR CLOCK COUNT FOR DATA WD
2783 012750 012705 026610           MOV       #INBUF,R5       ;GET STARTING ADDRESS FOR DATA BUFFER
2784 012754 162705 000002           SUB       #2,R5
2785 012760 012767 000025 166216       MOV       #21.,REPT1       ;SETUP COUNTER FOR 1ST SB BIT
2786 012766 012767 002200 166206       MOV       #1152.,REPT    ;SETUP COUNTER TO TRANSFER
2787                                   ;128 WORDS-9X128=1152
2788                                   ;2 CLOCKS PER 2 BITS OF DATA
2789 012774 104444                   15:     RBIT                   ;GET 2 DATA BITS
2790 012776 104440                   CLKR1                   ;CLOCK MR
2791 013000 104000                   HLT                    ;MR REG NOT CORRECT
2792 013002 104442                   CLKR2                   ;CLOCK MR REG
2793 013004 104000                   HLT                    ;MR REG NOT CORRECT
2794 013006 005367 166170           DEC       REPT           ;DONE WITH DATA BUFFER YET?
2795 013012 001370                   BNE       15            ;NO
  
```

2796	013014	032767	000400	166144	2S:	BIT	#BIT8,ONCEE	:DID WE ALREADY DO CRC?
2797	013022	001030				BNE	3S	:YES
2798	013024	052767	000400	166134		BIS	#BIT8,ONCEE	:NO SET CRC FLAG
2799	013032	016767	166146	166132		MOV	REPT1,SAVEE	:SAVE REPT1
2800	013040	004767	010412			JSR	PC,GENCRC	:GENERATE CRC WORD
2801								:AND LEAVE IN LOC "WORK"
2802	013044	012702	026610			MOV	#INBUF,R2	
2803	013050	016767	166116	166126		MOV	SAVEE,REPT1	:RESTORE REPT1
2804	013056	062702	000400			ADD	#400,R2	:STORE CRC WORD AT END OF
2805	013062	016712	166126			MOV	WORK,R2	:INBUF TABLE
2806	013066	010205				MOV	R2,R5	
2807	013070	162705	000002			SUB	#2,R5	
2808	013074	012767	000011	166100		MOV	#9.,REPT	:SETUP TO TRANSFER 1 WD
2809	013102	000734				BR	1S	:TRANSFER CRC WD
2810	013104	104422			3S:	MRCLK		:CLOCK MR REG
2811	013106	104420				MRCK		:CHECK MR REG
2812	013110	003601				3601		:TO EQUAL
2813	013112	104000				HLT		:3601
2814	013114	104422				MRCLK		:CLOCK MR REG
2815	013116	104420				MRCK		:CHECK MR
2816	013120	153601				153601		:TO EQUAL
2817	013122	104000				HLT		:153601
2818	013124	104422				MRCLK		:CLOCK MR REG
2819	013126	104420				MRCK		:CHECK MR
2820	013130	007601				7601		:TO EQUAL
2821	013132	104000				HLT		:7601
2822	013134	104422				MRCLK		:CLOCK MR REG
2823	013136	104420				MRCK		:CHECK MR
2824	013140	153601				153601		:TO EQUAL
2825	013142	104000				HLT		:153601
2826								
2827								:PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
2828								:STEP INTO END OF SECTOR DEAD BAND
2829								:EBL IS NOW ASSERTED
2830								
2831	013144	012767	000010	166030	MRD4:	MOV	#8.,REPT	
2832	013152	104422			1S:	MRCLK		:CLOCK MR REG
2833	013154	104420				MRCK		:CHECK MR REG
2834	013156	003601				3601		:TO EQUAL
2835	013160	104000				HLT		:3601
2836	013162	104422				MRCLK		:CLOCK MR REG
2837	013164	104420				MRCK		:CHECK MR
2838	013166	153601				153601		:REG TO
2839	013170	104000				HLT		:EQUAL 153601
2840	013172	005367	166004			DEC	REPT	:DONE YET?
2841	013176	001365				BNE	1S	:NO
2842								
2843								:PERFORM ONE MAINTENANCE CLOCK OPERATION
2844								:SHOULD GET STROBE BUFFER
2845								
2846	013200	104422				MRCLK		:CLOCK MR REG
2847	013202	104420				MRCK		:CHECK MR
2848	013204	007601				7601		:REG TO
2849	013206	104000				HLT		:EQUAL 7601

M05

```

2850      ;PERFORM ONE MAINTENANCE CLOCK OPERATION
2851      ;SHOULD COMPLETE TRANSFER.
2852
2853      013210  104422
2854      013212  022777  004270  165660  MRD5:  MRCLK      ;CLOCK MR REG
2855      013220  001401      CMP      #4270,RSCSI ;ANY ERRORS?
2856      013222  104054      BEQ      IS        ;NO
2857      013224  005777  165654  1S:    HLT      !DA!DS!MC
2858      013230  001401      TST      @RSMC     ;DID MC GO TO 0
2859      013232  104010      BEQ      +4        ;YES
2860      013234  022777  000001  165646  HLT      !MC       ;MC REG SHOULD=0
2861      013242  001401      CMP      #1,RSDA   ;DOES RSDA=1
2862      013244  104004      BEQ      +4        ;YES
2863      HLT      !DA   ;RSDA SHOULD=1
2864
2865      ;COMPARE DATA READ WITH INPUT BUFFER
2866      ;WILL ONLY TYPEOUT 10 ERRORS ---- BUT IF SW12 IS SET
2867      ;IT WILL TYPE OUT ALL ERRORS
2868
2869      013246  012700  026610  MRD6:  MOV      @INBUF,BAD ;GET STARTING LOC OF EXPECTED DATA
2870      013252  012701  027410      MOV      @OUTBUF,GOOD ;GET STARTING LOC OF DATA "READ" FROM DISK
2871      013254  012767  000012  165716  MOV      #12,REPT    ;SET UP ERROR COUNTER
2872      013264  012705  000201      MOV      #201,R5     ;COMPARE 1 SECTOR
2873      013270  005305  3S:    DEC      R5        ;DONE WITH SECTOR
2874      013272  001433      BEQ      2S        ;YES GET OUT
2875      013274  022021      CMP      (BAD)+,(GOOD)+ ;IS DATA CORRECT?
2876      013276  001774      BEQ      3S        ;YES
2877      013300  032777  010000  164262  BIT      @BIT12,@SWR ;TYPE ALL ERRORS?
2878      013306  001003      BNE      IS        ;YES
2879      013310  005367  165666  DEC      REPT       ;TYPED OUT 10 ERRORS YET?
2880      013314  001422      BEQ      2S        ;YES GET OUT
2881      013316  024041  1S:    CMP      -(BAD),-(GOOD) ;GET ERROR
2882      013320  104000      HLT      ;TYPE OUT ERROR
2883      013322  010067  165666  MOV      BAD,WORK   ;ASCIZ "BAD ADDRESS="
2884      013326  104402  013332  TYPE      +2        ;PUT WORK ON STACK
2885      013350  016746  165640  MOV      WORK,-(6)  ;TYPE STACK IN OCTAL - SUPRESS
2886      013354  104406      TYPES
2887      013356  022021      CMP      (BAD)+,(GOOD)+
2888      013360  000743      BR       3S
2889      2S:
2890      ;DONE
  
```

N05

MAINDEC-11-DERSD-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 66
 DERSDC.SRC 17-MAY-77 14:16 TST53 MAINTENANCE MODE DATA WRITE CHECK TEST

```

2889 :*****
2890 :TEST 53 MAINTENANCE MODE DATA WRITE CHECK TEST
2891 :*****
2892 013362 104400 TST53: SCOPE
2893
2894
2895 :MODULE TESTED: M7771, M7753, M7751
2896 :A ONE SECTOR TRANSFER IS DONE WITH A WRITE CHECK FUNCTION.
2897 :WITHIN THE RS04, A WRITE CHECK FUNCTION IS IDENTICAL TO A
2898 :READ FUNCTION.
2899 013364 104414 MRWCK: CLRDK ;CLEAR DRIVE REGISTERS
2900 013366 052767 000040 165572 BIS #BITS, ONCEE ;SET TYPE CLOCK COUNT FLAG
2901 013374 042767 147716 165564 BIC #147716, ONCEE ;CLEAR ALL OTHER FLAG BITS
2902 013402 104430 MRIND ;SEND INDEX PULSE TO MR REG
2903 013404 104420 MRCK ;CHECK MR REG
2904 013406 022701 ;TO EQUAL 22701
2905 013410 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
2906 ;BY SENDING 2 CLOCK PULSES
2907
2908 013412 012767 000004 165524 MOV #4, FLAG2 ;SET MC FLAG FOR CLKR1 ROUTINE
2909
2910 ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
2911 ;DATA BUFFER WORDS ARE :A WORD OF ALL 0'S
2912 ;A WORD OF ALL 1'S
2913 ;FLOATING 1'S PATTERN (16 WORDS)
2914 ;A PATTERN OF 146314 (110 WORDS)
2915
2916 013420 012702 026610 MOV #INBUF, R2 ;GET LOCATION OF INBUF
2917 013424 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
2918 013426 012722 177777 MOV #-1, (R2)+ ;2ND WORD OF ALL ONES
2919 013432 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
2920 013434 000261 SEC ;A PATTERN OF FLOATING ONES
2921 013436 006103 1S: ROL R3 ;GET PATTERN
2922 013440 103402 BCS 2S ;DONE GET OUT
2923 013442 010322 MOV R3, (R2)+ ;FILL BUFFER
2924 013444 000774 BR 1S ;CONT
2925 013446 012703 000156 2S: MOV #110, R3 ;FILL REMAINING PORTION OF
2926 013452 012704 146314 MOV #146314, R4 ;BUFFER WITH A PATTERN OF 146314
2927 013456 010422 3S: MOV R4, (R2)+ ;LOAD BUFFER
2928 013460 005303 DEC R3 ;DONE YET
2929 013462 001375 BNE 3S ;NO
2930
2931 ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
2932
2933 013464 012777 026610 165414 MOV #INBUF, #RSBA ;LOAD BUS ADDR REG
2934 013472 012777 177600 165404 MOV #177600, #RSWC ;LOAD WORD COUNT REG
2935 013500 012777 000051 165372 MOV #51, #RSCS1 ;LOAD WRITE CHECK COMMAND
2936 013506 104446 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
2937 ;TO CLEAR OUT COUNTERS AND REGISTERS
2938 ;THAT OTHERWISE COULD NOT BE CLEARED.
2939 013510 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
2940 013512 104450 SPASS ;CLOCK MR SECTOR PULSE = 1
    
```

```

2941 ;ASSERT INDEX PULSE TO INITIALIZE THE DRIVE
2942 013514 104430 MRIND
2943 013516 104420 MRCK ;CHECK MR REG TO EQUAL
2944 013520 022701 22701 ;22701 FOR A
2945 013522 104000 HLT ;WRITE CHECK COMMAND
2946
2947 ;STEP THRU RESYNC PERIOD
2948
2949 013524 012767 001000 165450 MOV #512.,REPT
2950 013532 052767 000040 165426 MRWCK1: MRCK #BITS,ONCEE ;TYPE OUT CLOCK COUNT IF ERRORS OCCUR
2951 013540 104422 MRCLK ;CLOCK MR REG
2952 013542 104420 MRCK ;CHECK FOR
2953 013544 072701 72701 ;CORRECT DATA
2954 013546 104000 HLT ;MR=BAD GOOD=CORRECT DATA
2955 013550 104422 MRCLK ;CLOCK MR REG
2956 013552 104420 MRCK ;CHECK FOR
2957 013554 022701 22701 ;CORRECT DATA
2958 013556 104000 HLT ;ERROR WHILE CLOCKING THROUGH RESYNC
2959 013560 005367 165416 DEC REPT ;FINISH LOOPING
2960 013564 001365 BNE MRWCK1 ;THROUGH RESYNC PERIOD
2961
2962 ;ONE MORE CLOCK PULSE SHOULD ASSERT SECTOR PULSE
2963 ;SP=0 EQUALS SECTOR PULSE
2964 013566 104422 MRCLK ;CLOCK MR REG
2965 013570 104420 MRCK ;MR SHOULD
2966 013572 072301 72301 ;EQUAL 72301
2967 013574 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2968 013576 104422 MRCLK ;CLOCK MR REG
2969 013600 104420 MRCK ;CHECK MR
2970 013602 022301 22301 ;TO EQUAL 22301
2971 013604 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2972
2973 ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
2974
2975 013606 012767 000107 165366 MRWCK2: MOV #71.,REPT
2976 013614 104422 MRCLK ;CLOCK MR REG
2977 013616 104420 MRCK ;CHECK MR REG
2978 013620 073701 73701 ;TO EQUAL 73701
2979 013622 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2980 013624 104422 MRCLK ;CLOCK MR REG
2981 013626 104420 MRCK ;CHECK MR REG
2982 013630 023701 23701 ;TO EQUAL 23701
2983 013632 104000 HLT ;MR=BAD GOOD=CORRECT ANS
2984 013634 005367 165342 DEC REPT ;DONE YET
2985 013640 001365 BNE MRWCK2 ;NO LOOP
2986 013642 104422 MRCLK ;CLOCK MR REG
2987 013644 104420 MRCK ;CHECK MR REG
2988 013646 073701 73701 ;TO EQUAL 73701
2989 013650 104000 HLT ;MR=BAD GOOD=CORRECT ANS
    
```

```

2990 ;READ SYNC"1"
2991
2992 013652 012777 000055 165244 MOV #55,DRSMR
2993 013660 012777 000045 165236 MOV #45,DRSMR
2994 013666 104420 MRCK ;CHECK MR FOR
2995 013670 023745 23745 ;FOR CORRECT
2996 013672 104000 HLT ;ANS IS IN GOOD
2997 013674 012777 000055 165222 MOV #55,DRSMR
2998 013702 012777 000045 165214 MOV #45,DRSMR
2999 013710 104420 MRCK
3000 013712 173745 173745
3001 013714 104000 HLT
3002
3003 ;READ DATA
3004 013716 005067 165302 MRWCK3: CLR WORK3 ;CLEAR CLOCK COUNT FOR DATA WD
3005 013722 012705 026610 MOV #INBUF,R5 ;GET STARTING ADDRESS FOR DATA BUFFER
3006 013726 162705 000002 SUB #2,R5
3007 013732 012767 000025 165244 MOV #21,REPT1 ;SETUP COUNTER FOR 1ST SB BIT
3008 013740 012767 002200 165234 MOV #1152,REPT ;SETUP COUNTER TO TRANSFER
3009 ;128 WORDS-9X128=1152
3010 ;2 CLOCKS PER 2 BITS OF DATA
3011 013746 104444 1S: RBIT ;GET 2 DATA BITS
3012 013750 104440 CLKR1 ;CLOCK MR
3013 013752 104000 HLT ;MR REG NOT CORRECT
3014 013754 104442 CLKR2 ;CLOCK MR REG
3015 013756 104000 HLT ;MR REG NOT CORRECT
3016 013760 005367 165216 DEC REPT ;DONE WITH DATA BUFFER YET?
3017 013764 001370 BNE 1S ;NO
3018 013766 032767 000400 165172 2S: BIT #BIT8,ONCEE ;DID WE ALREADY DO CRC?
3019 013774 001030 BNE 3S ;YES
3020 013776 052767 000400 165162 BIS #BIT8,ONCEE ;NO SET CRC FLAG
3021 014004 016767 165174 165160 MOV REPT1,SAVEE ;SAVE REPT1
3022 014012 004767 007440 JSR PC,GENCRC ;GENERATE CRC WORD
3023 ;AND LEAVE IN LOC "WORK"
3024 014016 012702 026610 MOV #INBUF,R2 ;RESTORE REPT1
3025 014022 016767 165144 165154 MOV SAVEE,REPT1 ;STORE CRC WORD TO BE READ
3026 014030 062702 000400 ADD #400,R2 ;AT END OF INBUF TABLE
3027 014034 016712 165154 MOV WORK,DR2
3028 014040 010205 MOV R2,R5
3029 014042 162705 000002 SUB #2,R5
3030 014046 012767 000011 165126 MOV #9,REPT ;SETUP TO TRANSFER 1 WD
3031 014054 000734 BR 1S ;TRANSFER CRC WD
    
```

```

3032 014056 104422          3S:  MRCLK          :CLOCK MR REG
3033 014060 104420          MRCK          :CHECK MR REG
3034 014062 003701          3701          :TO EQUAL
3035 014064 104000          HLT          :3701
3036 014066 104422          MRCLK          :CLOCK MR REG
3037 014070 104420          MRCK          :CHECK MR
3038 014072 153701          153701        :TO EQUAL
3039 014074 104000          HLT          :153701
3040 014076 104422          MRCLK          :CLOCK MR REG
3041 014100 104420          MRCK          :CHECK MR
3042 014102 007701          7701          :TO EQUAL
3043 014104 104000          HLT          :7701
3044 014106 104422          MRCLK          :CLOCK MR REG
3045 014110 104420          MRCK          :CHECK MR
3046 014112 153701          153701        :TO EQUAL
3047 014114 104000          HLT          :153701
3048
3049          :PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3050          :STEP INTO END OF SECTOR DEAD BAND
3051          :EBL IS NOW ASSERTED
3052
3053 014116 012767 000010 165056 MRWCK4: MOV      #8.,REPT
3054 014124 104422          1S:  MRCLK          :CLOCK MR REG
3055 014126 104420          MRCK          :CHECK MR REG
3056 014130 003701          3701          :TO EQUAL
3057 014132 104000          HLT          :3601
3058 014134 104422          MRCLK          :CLOCK MR REG
3059 014136 104420          MRCK          :CHECK MR
3060 014140 153701          153701        :REG TO
3061 014142 104000          HLT          :EQUAL 153601
3062 014144 005367 165032  DEC      REPT
3063 014150 001365          BNE      1S
3064          :NO
3065          :PERFORM ONE MAINTENANCE CLOCK OPERATION
3066          :SHOULD GET STROBE BUFFER
3067
3068 014152 104422          MRCLK          :CLOCK MR REG
3069 014154 104420          MRCK          :CHECK MR
3070 014156 007701          7701          :REG TO
3071 014160 104000          HLT          :EQUAL 7601
3072
3073          :PERFORM ONE MAINTENANCE CLOCK OPERATION
3074          :SHOULD COMPLETE TRANSFER.
3075
3076 014162 104422          MRWCK5: MRCLK          :CLOCK MR REG
3077 014164 022777 004250 164706  CMP      #4250,RSRCS1 :ANY ERRORS?
3078 014172 001401          BEQ      1S
3079 014174 104054          HLT      !DA!DS!WC :NO
3080 014176 005777 164702  1S:  TST      RSWC
3081 014202 001401          BEQ      +4
3082 014204 104010          HLT      !WC
3083 014206 022777 000001 164674  CMP      #1,RSRSDA
3084 014214 001401          BEQ      +4
3085 014216 104004          HLT      !DA
          :DID WC GO TO 0
          :YES
          :WC REG SHOULD=0
          :DOES RSDA=1
          :YES
          :RSDA SHOULD=1
    
```

E06

3086
3087
3088
3089 014220 104400
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099

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

TST54: SCOPE

3100 014222 012767 000040 164714
3101 014230 104414
3102 014232 052767 000040 164726
3103 014240 042767 147716 164720
3104 014246 104430
3105 014250 104420
3106 014252 022701
3107 014254 104424
3108
3109 014256 032767 000020 164660
3110 014264 001023
3111 014266 012767 000001 164676
3112
3113
3114
3115
3116
3117
3118
3119
3120
3121
3122
3123
3124
3125

: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
CLDK ;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.

3126 014274 012702 026610
3127 014300 012703 000016
3128 014304 005022
3129 014306 005303
3130 014310 001375
3131 014312 012722 000012
3132 014316 012722 020000
3133 014322 012703 000200
3134 014326 005022
3135 014330 005303
3136 014332 001375

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
MOV #128,R3 ;SETUP COUNTER
2\$: CLR (R2)+ ;TO CLEAR THE
DEC R3 ;REMAINING WORDS
BNE 2\$;FOR THAT SECTOR

G06

```

3185 ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3186
3187 014502 012767 000107 164472 MRCRC2: MOV #71.,REPT
3188 014510 104422 MRCLK ;CLOCK MR REG
3189 014512 104420 MRCK ;CHECK MR REG
3190 014514 073601 73601 ;TO EQUAL 73601
3191 014516 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3192 014520 104422 MRCLK ;CLOCK MR REG
3193 014522 104420 MRCK ;CHECK MR REG
3194 014524 023601 23601 ;TO EQUAL 23601
3195 014526 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3196 014530 005367 164446 DEC REPT ;DONE YET
3197 014534 001365 BNE MRCRC2 ;NO LOOP
3198 014536 104422 MRCLK ;CLOCK MR REG
3199 014540 104420 MRCK ;CHECK MR REG
3200 014542 073601 73601 ;TO EQUAL 73601
3201 014544 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3202
3203 ;READ SYNC"1"
3204
3205 014546 012777 000055 164350 MOV #55,RSMR
3206 014554 012777 000045 164342 MOV #45,RSMR
3207 014562 104420 MRCK ;CHECK MR REG
3208 014564 023645 23645 ;FOR CORRECT
3209 014566 104000 HLT ;ANS IS IN GOOD
3210 014570 012777 000055 164326 MOV #55,RSMR
3211 014576 012777 000045 164320 MOV #45,RSMR
3212 014604 104420 MRCK
3213 014606 173645 173645
3214 014610 104000 HLT
3215
3216 ;READ DATA
3217 014612 005067 164406 MRCRC3: CLR WORK3 ;CLEAR CLOCK COUNT FOR DATA WD
3218 014616 012705 026610 MOV #INBUF,R5 ;GET STARTING ADDRESS FOR DATA BUFFER
3219 014622 162705 000002 SUB #2,R5
3220 014626 012767 000025 164350 MOV #21,REPT1 ;SETUP COUNTER FOR 1ST SB BIT
3221 014634 012767 002200 164340 MOV #1152.,REPT ;SETUP COUNTER TO TRANSFER
3222 ;128 WORDS-9X128=1152
3223 ;2 CLOCKS PER 2 BITS OF DATA
3224 014642 104444 1S: RBIT ;GET 2 DATA BITS
3225 014644 104440 CLKR1 ;CLOCK MR
3226 014646 104000 HLT ;MR REG NOT CORRECT
3227 014650 104442 CLKR2 ;CLOCK MR REG
3228 014652 104000 HLT ;MR REG NOT CORRECT
3229 014654 005367 164322 DEC REPT ;DONE WITH DATA BUFFER YET?
3230 014660 001370 BNE 1S ;NO

```


H06

MAINDEC-11-DERSD-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 73
 DERSDC.SRC 17-MAY-77 14:16 TST54 MAINTENANCE MODE CRC TEST 1 (NO DCK ERRORS)

3231	014662	032767	000400	164276	2S:	BIT	#BIT8,ONCEE	:DID WE ALREADY DO CRC?
3232	014670	001020				BNE	3S	:YES
3233	014672	052767	000400	164266		BIS	#BIT8,ONCEE	:NO SET CRC FLAG
3234	014700	012702	026610			MOV	#INBUF,R2	:MOVE CRC
3235	014704	062702	000440			ADD	#440,R2	:WORD TO END OF
3236	014710	016712	164256		4S:	MOV	SAVEE,#R2	:INBUF TABLE
3237	014714	010205			5S:	MOV	R2,R5	:GET CRC WORD
3238	014716	162705	000002			SUB	#2,R5	
3239	014722	012767	000011	164252		MOV	#9.,REPT	:SETUP TO TRANSFER 1 WD
3240	014730	000744				BR	1S	:TRANSFER CRC WD
3241	014732	104422			3S:	MRCLK		:CLOCK MR REG
3242	014734	104420				MRCK		:CHECK MR REG
3243	014736	003601				3601		:TO EQUAL
3244	014740	104000				HLT	3601	
3245	014742	104422				MRCLK		:CLOCK MR REG
3246	014744	104420				MRCK		:CHECK MR
3247	014746	153601				153601		:TO EQUAL
3248	014750	104000				HLT	153601	
3249	014752	104422				MRCLK		:CLOCK MR REG
3250	014754	104420				MRCK		:CHECK MR
3251	014756	007601				7601		:TO EQUAL
3252	014760	104000				HLT	7601	
3253	014762	104422				MRCLK		:CLOCK MR REG
3254	014764	104420				MRCK		:CHECK MR
3255	014766	153601				153601		:TO EQUAL
3256	014770	104000				HLT	153601	

```

3257 ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3258 ;STFP INTO END OF SECTOR DEAD BAND
3259 ;EPL IS NOW ASSERTED
3260
3261 014772 012767 000010 164202 MRCRC4: MOV      #8.,REPT
3262
3263 015000 104422          1$:  MRCLK          ;CLOCK MR REG
3264 015002 104420          MRCK           ;CHECK MR REG
3265 015004 003601          3601           ;TO EQUAL
3266 015006 104000          HLT            ;3601
3267 015010 104422          MRCLK          ;CLOCK MR REG
3268 015012 104420          MRCK           ;CHECK MR
3269 015014 153601          153601        ;REG TO
3270 015016 104000          HLT            ;EQUAL 153601
3271 015020 005367 164156  DEC           ;DONE YET?
3272 015024 001365          BNE          1$          ;NO
3273
3274 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3275 ;SHOULD GET STROBE BUFFER
3276
3277 015026 104422          MRCLK          ;CLOCK MR REG
3278 015030 104420          MRCK           ;CHECK MR
3279 015032 007601          7601           ;REG TO
3280 015034 104000          HLT            ;EQUAL 7601
3281
3282 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3283 ;SHOULD COMPLETE TRANSFER.
3284
3285 015036 104422          MRCRC5: MRCLK        ;CLOCK MR REG
3286 015040 022777 004270 164032  CMP          #4270,@RSCS1 ;ANY ERRORS?
3287 015046 001401          BEQ          1$          ;NO
3288 015050 104054          HLT          !DA!DS!WC
3289 015052 005777 164026  1$:  TST          @R5WC    ;DID WC GO TO 0
3290 015056 001401          BEQ          +4          ;YES
3291 015060 104010          HLT          !WC      ;WC REG SHOULD=0
3292 015062 006167 164104  ROL          SAVEE     ;GET NEXT CRC WORD
3293 015066 103404          BCS          2$          ;DONE - BRANCH
3294 015070 004767 010506  JSR          PC,MDATA  ;SHIFT DATA PATTERN
3295 015074 000167 177130  JMP          MRCRC     ;RESTART TEST WITH NEW DATA PATTERN
3296 015100          2$:  ;DONE
    
```

J06

```

3297 :*****
3298 :TEST 55 MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)
3299 :*****
3300 015100 104400
3301 :TST55: SCOPE
3302 :MODULE TESTED M7753
3303 :THIS TEST IS SIMILAR TO CRC TEST 1 EXCEPT THAT THE DATA
3304 :PATTERN HAS BEEN MODIFIED TO LEAVE A SINGLE BIT SET IN THE
3305 :CRC REGISTER AFTER BOTH DATA AND CRC WORDS HAVE BEEN "READ".
3306 :THIS CAUSES A DCK ERROR. THE READ SEQUENCE IS REPEATED 16
3307 :TIMES TO TEST THAT EACH BIT IN THE CRC REGISTER CAN CAUSE A DCK ERROR.
3307 015102 012767 000040 164034 :MOV #40,FLAG2 ;CLEAR TST FLAG
3308 015110 104414 :MRDCK: CLDK ;CLEAR DRIVE REGISTERS
3309 015112 052767 000040 164046 :BIS #BITS,ONCEE ;TYPE CLOCK COUNT IF ERROR OCCURS
3310 015120 042767 147716 164040 :BIC #147716,ONCEE ;CLEAR ALL OTHER FLAG BITS
3311 015126 104430 :MRIND ;SEND INDEX PULSE TO MR REG
3312 015130 104420 :MRCK ;CHECK MR REG
3313 015132 022701 :22701 ;TO EQUAL 22701
3314 015134 104424 :MRINT ;INIT MAINT MODE (CLEAR MRSP)
3315 : ;BY SENDING 2 CLOCK PULSES
3316 015136 032767 000020 164000 :BIT #BIT4,FLAG2 ;FIRST TIME THROUGH
3317 015144 001023 :BNE 35 ;NO
3318 015146 012767 000001 164016 :MOV #1,SAVEE ;LOAD 1ST CRC WORD
3319 : ;FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR) CREATE BUFFER
3320 : ;WITH 144 WORDS OF 16 BITS WHICH = THE NO. OF BITS IN 128 18 BIT WORDS
3321 : ;DATA BUFFER CONTAINS 15 WORDS OF ZEROS
3322 : ; A WORD OF 1
3323 : ; A WORD OF 42000
3324 : ; 127 WORDS OF ZEROS
3325 015154 012702 026610 :MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
3326 015160 012703 000017 :MOV #15,R3 ;SETUP COUNTER
3327 015164 005022 15: CLR (R2)+ ;TO CLEAR THE
3328 015166 005303 :DEC R3 ;FIRST 15
3329 015170 001375 :BNE 15 ;WORDS
3330 015172 012722 000001 :MOV #1,(R2)+ ;LOAD A 1
3331 015176 012722 042000 :MOV #42000,(R2)+ ;LOAD A 42000
3332 015202 012703 000177 :MOV #127,R3 ;SETUP COUNTER
3333 015206 005022 25: CLR (R2)+ ;TO CLEAR THE
3334 015210 005303 :DEC R3 ;REMAINING WORDS
3335 015212 001375 :BNE 25 ;FOR THAT SECTOR
3336 : ;SETUP CONTROLLER TO TRANSFER 128 WORDS OF DATA (1 SECTOR) FROM SECTOR 0
3337 015214 012777 027410 163664 35: MOV #OUTBUF,ARSA ;LOAD BUS ADDR REG
3338 015222 012777 177600 163654 :MOV #177600,ARSMC ;LOAD WORD COUNT REG
3339 015230 012777 000071 163642 :MOV #71,ARSCS1 ;LOAD READ COMMAND
3340 015236 012702 000200 :MOV #200,R2
3341 015242 012703 027410 :MOV #OUTBUF,R3
3342 015246 052767 000020 163670 :BIS #BIT4,FLAG2 ;SET 1ST TIME THROUGH FLAG
3343 015254 005023 45: CLR (R3)+
3344 015256 005302 :DEC R2
3345 015260 001375 :BNE 45
3346 015262 104446 :GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3347 : ;TO CLEAR OUT COUNTERS AND REGISTERS
3348 : ;THAT OTHERWISE COULD NOT BE CLEARED.
3349 015264 104220 :HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3350 015266 104450 :SPASS ;CLOCK MR REG SP = 1

```



```

3382 ;PERFORM 71 DOUBLE MAINT CLOCK OPERATIONS--
3383
3384 015362 012767 000107 163612 MRDCK2: MOV #71.,REPT
3385 015370 104422 MRCLK ;CLOCK MR REG
3386 015372 104420 MRCK ;CHECK MR REG
3387 015374 073601 73601 ;TO EQUAL 73601
3388 015376 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3389 015400 104422 MRCLK ;CLOCK MR REG
3390 015402 104420 MRCK ;CHECK MR REG
3391 015404 023601 23601 ;TO EQUAL 23601
3392 015406 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3393 015410 005367 163566 DEC REPT ;DONE YET
3394 015414 001365 BNE MRDCK2 ;NO LOOP
3395 015416 104422 MRCLK ;CLOCK MR REG
3396 015420 104420 MRCK ;CHECK MR REG
3397 015422 073601 73601 ;TO EQUAL 73601
3398 015424 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3399
3400 ;READ SYNC"1"
3401
3402 015426 012777 000055 163470 MOV #55,RSMR
3403 015434 012777 000045 163462 MOV #45,RSMR
3404 015442 104420 MRCK ;CHECK MR REG
3405 015444 023645 23645 ;TO EQUAL
3406 015446 104000 HLT ;CORRECT ANS IN GOOD
3407 015450 012777 000055 163446 MOV #55,RSMR
3408 015456 012777 000045 163440 MOV #45,RSMR
3409 015464 104420 MRCK
3410 015466 173645 173645
3411 015470 104000 HLT
3412
3413 ;READ DATA
3414 015472 005067 163526 MRDCK3: CLR WORK3 ;CLEAR CLOCK COUNT FOR DATA WD
3415 015476 012705 026610 MOV #INBUF,RS ;GET STARTING ADDRESS FOR DATA BUFFER
3416 015502 162705 000002 SUB #2,RS
3417 015506 012767 000025 163470 MOV #21,REPT1 ;SETUP COUNTER FOR 1ST SB BIT
3418 015514 012767 002200 163460 MOV #1152.,REPT ;SETUP COUNTER TO TRANSFER
3419 ;128 WORDS-9X128=1152
3420 ;2 CLOCKS PER 2 BITS OF DATA
3421 015522 104444 15: RBIT ;GET 2 DATA BITS
3422 015524 104440 CLKR1 ;CLOCK MR
3423 015526 104000 HLT ;MR REG NOT CORRECT
3424 015530 104442 CLKR2 ;CLOCK MR REG
3425 015532 104000 HLT ;MR REG NOT CORRECT
3426 015534 005367 163442 DEC REPT ;DONE WITH DATA BUFFER YET?
3427 015540 001370 BNE 15 ;NO
    
```

M06

MAINDEC-11-DERSO-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 78
 DERSDC.SRC 17-MAY-77 14:16 TST55 MAINTENANCE MODE CRC TEST 2 (CAUSE DCK ERRORS)

3428	015542	032767	000400	163416	2S:	BIT	#BIT8,ONCEE	:DID WE ALREADY DO CRC?
3429	015550	001020				BNE	3S	:YES
3430	015552	052767	000400	163406		BIS	#BIT8,ONCEE	:NO SET CRC FLAG
3431	015560	012702	026610			MOV	#INBUF,R2	:MOVE CRC
3432	015564	062702	000440			ADD	#440,R2	:WORD TO END OF
3433	015570	012712	000000		4S:	MOV	#0,R2	:INBUF TABLE
3434	015574	010205			5S:	MOV	R2,R5	:GET CRC WORD
3435	015576	162705	000002			SUB	#2,R5	
3436	015602	012767	000011	163372		MOV	#9.,REPT	:SETUP TO TRANSFER 1 WD
3437	015610	000744				BR	1S	:TRANSFER CRC WD
3438	015612	104422			3S:	MRCLK		:CLOCK MR REG
3439	015614	104420				MRCK		:CHECK MR REG
3440	015616	003601				3601		:TO EQUAL
3441	015620	104000				HLT	3601	
3442	015622	104422				MRCLK		:CLOCK MR REG
3443	015624	104420				MRCK		:CHECK MR
3444	015626	153601				153601		:TO EQUAL
3445	015630	104000				HLT	153601	
3446	015632	104422				MRCLK		:CLOCK MR REG
3447	015634	104420				MRCK		:CHECK MR
3448	015636	007601				7601		:TO EQUAL
3449	015640	104000				HLT	7601	
3450	015642	104422				MRCLK		:CLOCK MR REG
3451	015644	104420				MRCK		:CHECK MR
3452	015646	153601				153601		:TO EQUAL
3453	015650	104000				HLT	153601	

N06

```

3454 ;PERFORM 8 DOUBLE MAINTENANCE CLOCK OPERATIONS
3455 ;STEP INTO END OF SECTOR DEAD BAND
3456 ;EBL IS NOW ASSERTED
3457
3458 015652 012767 000010 163322 MRDCK4: MOV #8.,REPT
3459
3460 015660 104422 1S: MRCLK ;CLOCK MR REG
3461 015662 104420 MRCK ;CHECK MR REG
3462 015664 003601 3601 ;TO EQUAL
3463 015666 104000 HLT ;3601
3464 015670 104422 MRCLK ;CLOCK MR REG
3465 015672 104420 MRCK ;CHECK MR
3466 015674 153601 153601 ;REG TO
3467 015676 104000 HLT ;EQUAL 153601
3468 015700 005367 163276 DEC REPT ;DONE YET?
3469 015704 001365 BNE 1S ;NO
3470
3471 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3472 ;SHOULD GET STROBE BUFFER
3473
3474 015706 104422 MRCLK ;CLOCK MR REG
3475 015710 104420 MRCK ;CHECK MR
3476 015712 007601 7601 ;REG TO
3477 015714 104000 HLT ;EQUAL 7601
3478
3479 ;PERFORM ONE MAINTENANCE CLOCK OPERATION
3480 ;SHOULD COMPLETE TRANSFER.
3481
3482 015716 104422 MRDCK5: MRCLK ;CLOCK MR REG
3483 015720 022777 144270 163152 CMP #144270,RSRCS1 ;IS RSCS1 CORRECT?
3484 015726 001401 BEQ 1S ;YES
3485 015730 104054 HLT !DA!DS!WC ;
3486 015732 005777 163146 1S: TST RSWC ;DID WC GO TO 0
3487 015736 001401 BEQ .+4 ;YES
3488 015740 104010 HLT !WC ;WC REG SHOULD=0
3489 015742 022777 100000 163144 CMP #100000,RSER ;DID DCK SET?
3490 015750 001417 BEQ 3S ;YES
3491 015752 104050 HLT !DS!WC ;
3492 015754 104402 015760 TYPE .+2 ;ASCIZ <15><12>"DCK DID NOT SET "
3493 016004 004767 004210 JSR PC,CRCTYP ;GET IC THAT FAILED AND TYPE IT
3494 016010 000241 3S: CLC ;
3495 016012 006167 163154 ROL SAVEE ;GET NEXT CRC WORD
3496 016016 103404 BCS 2S ;DONE - BRANCH
3497 016020 004767 007556 JSR PC,MDATA ;SHIFT DATA PATTERN
3498 016024 000167 177060 JMP MRDCK ;RESTART TEST WITH NEW DATA PATTERN
3499 016030 2S: ;DONE
  
```

3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544

016030 104400

016032 104414
016034 104430
016036 104420
016040 022701
016042 104424
016044 012777 177777 163042
016052 016777 163104 163036
016060 012777 027410 163020
016066 012777 177777 163010
016074 012777 000071 162776
016102 032777 000001 162770
016110 001401
016112 104140
016114 012767 177777 163072 1S:
016122 005367 163066 5S:
016126 000240
016130 000240
016132 001373
016134 017700 162742
016140 012701 001100
016144 056701 163010
016150 020001
016152 001401
016154 104000

:TEST 56 IGNORE F I G N TEST

TST56: SCOPE

:MODULE TESTED: M7759, M7770
:PUT THE DISK 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 IS TURN SHOULD
:CAUSE "TRE" AND "SC" TO SET IN RSCS1.

MRIFT: CLRK ;CLEAR ALL REGISTERS
MRIND ;SEND INDEX PULSE TO MR REG
MRCK ;CHECK MR REG
22701 ;TO EQUAL 22701
MRINT ;INIT MAINT MODE (CLEAR MRSP)
MOV #1,RSER ;SET ERRORS
MOV UNITSV,RSAS ;CLEAR ATA BIT IN RSAS
 ;AND ERROR BITS IN RSCS1
MOV #OUTBUF,RSBA ;LOAD RSBA
MOV #1,RSAC ;LOAD RSMC
MOV #71,RSXS1 ;LOAD READ FUNCTION
BIT #BIT0,RSXS1 ;IS "GO" BIT ZERO?
BEQ IS ;YES
HLT !DS!AS ;"GO" BIT IN RSCS1 SHOULD NOT
 ;LOAD IF ERRORS ARE PRESENT IN THE DRIVE
 ;SETUP TIMEOUT FOR MXF ERROR
1S: MOV #177777,WORK
5S: DEC WORK
NOP
NOP
BNE 5S
MOV #RSXS2,BAD ;CHECK RSCS2 FOR MXF
MOV #1100,GOOD ;GET CORRECT ANS
BIS UNNUM,GOOD ;FOR RSCS2
CMP BAD,GOOD ;IS RSCS2 CORRECT
BEQ 2S ;YES
 ;BAD=RSCS2 GOOD=CORRECT ANS
 ;MXF SHOULD BE SET IN RSCS2
 ;FOR A READ WAS ISSUED
 ;WITH ERROR BITS SET IN RSER.
2S: CMP #144270,RSXS1 ;IS RSCS1 CORRECT?
BEQ 3S ;YES
HLT !DS!ER ;SC AND TRE SHOULD BE SET FOR
 ;MXF SHOULD BE SET IN RSCS2


```

3569 :*****
3570 :TEST 57 INVALID ADDRESS ERROR (IAE) TEST
3571 :*****
3572 016250 104400 †TST57: SCOPE
3573
3574 :MODULE TESTED M7754, M7770
3575 :FLOAT A 1 THROUGH THE FOUR SPARE ADDRESS BITS IN THE DISK
3576 :ADDRESS REGISTER (RSDA). THIS SHOULD CAUSE "IAE" TO SET IN
3577 :THE ERROR REGISTER (RSER) WHEN A READ FUNCTION IS LOADED INTO
3578 :RSCS1 WHICH IN TURN SHOULD CAUSE ATTENTION TO SET IN THE
3579 :DRIVE STATUS REGISTER (RSDS) AND "TR" AND "SC" TO SET IN THE
3580 :CONTROL REGISTER (RSCS1).
3581 016252 042767 000040 162706 BIC #BIT5,ONCEE ;CLEAR CLK CNT FLAG
3582 016260 012702 004000 MOV #4000,R2 ;LOAD R2 WITH INVALID ADDR
3583 016264 012767 016272 162516 MOV #45,LAD ;LOOP TO HERE ON ERROR
3584 016272 104416 4S: MROMD ;PUT DRIVE IN MAINT MODE
3585 016274 104420 MRCK ;CHECK MAINT REG
3586 016276 022701 22701
3587 016300 104424 MRINT
3588 016302 032767 000004 162656 BIT #BIT2,ONCEE ;INIT MAINT MODE (CLEAR MRSP)
3589 016310 001002 BNE 15 ;LOOPING ON ERRORS)
3590 016312 006102 ROL R2 ;YES
3591 016314 103454 BCS IADONE ;GET INVALID ADDRESS
3592 016316 010277 162566 1S: MOV R2,RSDA ;DONE FLOATING A ONE YET?
3593 016322 012777 000071 162550 MOV #71,RSCS1 ;LOAD RSDA WITH INVALID ADDRESS
3594 016330 022777 002000 162556 CMP #2000,RSER ;DO A READ TO INVALID ADDR
3595 016336 001404 BEQ 2S ;IS RSER CORRECT?
3596 016340 052767 000004 162620 BIS #BIT2,ONCEE ;YES
3597 016346 104044 HLT !DS!DA ;SET ERROR BIT
3598 ;RSER SHOULD=2000 FOR
3599 ;A READ COMMAND WAS GIVEN
3600 016350 042767 000004 162610 2S: BIC #BIT2,ONCEE ;TO AN ILLEGAL ADDRESS
3601 016356 022777 150600 162526 CMP #150600,RSDS ;CLEAR ERROR FLAG
3602 016364 001404 BEQ 3S ;DID IAE SET?
3603 016366 052767 000004 162572 BIS #BIT2,ONCEE ;YES
3604 016374 104044 HLT !DS!DA ;SET ERROR BIT
3605 ;RSDS SHOULD=150600 FOR
3606 016376 042767 000004 162562 3S: BIC #BIT2,ONCEE ;IAE SHOULD BE SET IN RSER
3607 016404 032777 100000 162466 BIT #BIT15,RSCS1 ;CLEAR ERROR FLAG
3608 016412 001004 BNE 5S ;DID SC SET?
3609 016414 052767 000004 162544 BIS #BIT2,ONCEE ;YES
3610 016422 104044 HLT !DA!DS ;SET ERROR BIT
3611 ;SC SHOULD BE SET IN RSCS1
3612 016424 042767 000004 162534 5S: BIC #BIT2,ONCEE ;FOR IAE SHOULD BE SET IN RSER
3613 016432 104414 CLRDK ;CLEAR ERROR BIT
3614 016434 005777 162454 TST RSER ;CLEAR ALL ERRORS
3615 016440 001401 BEQ +4 ;DID IAE CLEAR?
3616 016442 104040 HLT !DS ;YES
3617 016444 000712 BR 4S ;IAE DID NOT CLEAR
3618 016446 IADONE: ;CONTINUE
;DONE

```

E07

```

3619 :*****
3620 :TEST 60 OPERATION INCOMPLETE ERROR TEST
3621 :*****
3622 016446 104400 TST60: SCOPE
3623
3624 ;MODULE TESTED M7770
3625 ;PUT THE DISK IN MAINTENANCE MODE AND START A READ COMMAND
3626 ;THEN ISSUE THREE DISK "INDEX" PULSES TO SIMULATE A COMPLETE
3627 ;ROTATION OF THE DISK SURFACE. THE THIRD INDEX PULSE SHOULD
3628 ;CAUSE OPERATION IN COMPLETE "OPI" TO APPEAR IN THE DRIVE ERROR
3629 ;REGISTER (RSER) AND "ATA" AND "ERR" IN THE DRIVE STATUS REGISTER (RSDS)
3630
3631 016450 104414 MROPI: CLDK ;CLEAR ALL DRIVE REGISTERS
3632 016452 013777 027410 162426 MOV @#OUTBUF,@RSBA ;SETUP RSBA
3633 016460 012777 177777 162416 MOV #-1,@RSCW ;SETUP RSWC
3634
3635 016466 104430 MRIND ;SEND INDEX PULSE TO MR REG
3636 016470 104420 MRCK ;CHECK MAINT REG
3637 016472 022701 22701 ;TO EQUAL 22701
3638 016474 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3639
3640 016476 012777 000071 162374 MOV #71,@RSCS1 ;LOAD A READ COMMAND
3641
3642 016504 104430 MRIND ;ISSUE THREE INDEX
3643 016506 104430 MRIND ;PULSES TO
3644 016510 104430 MRIND ;CAUSE OPI
3645
3646 ;NOW CHECK FOR CORRECT ERRORS IN RSER AND RSDS
3647 016512 017700 162376 MOV @RSER,BAD ;GET RSER
3648 016516 012701 020000 MOV #20000,GOOD ;GET CORRECT ANS
3649 016522 020100 CMP GOOD,BAD ;DID OPI SET IN RSER?
3650 016524 001434 BEQ 1$ ;YES
3651 016526 104402 016532 TYPE ,.+2 ;.ASCIZ <15><12>"OPI IN RSER SHOULD SET-3 INDEX PULSES W
3652 016614 104000 HLT ;RSER=BAD GOOD=CORRECT ANS
3653
3654 016616 022777 150600 162266 1$: CMP #150600,@RSDS ;DID CORRECT ERRORS SET?
3655 016624 001401 BEQ 2$ ;YES
3656 016626 104040 HLT !DS ;RSDS SHOULD=150600 BECAUSE
3657 ;OF OPI ERROR IN RSER
3658 016630 022777 144270 162242 2$: CMP #144270,@RSCS1 ;DID SC AND TRE SET IN RSCS1?
3659 016636 001401 BEQ MROPIA ;YES
3660 016640 104050 HLT !DS!WC ;SC AND TRE SHOULD SET IN RSCS1
3661 ;BECAUSE OF ERROR IN RSER
3662 016642 104414 MROPIA: CLDK ;CLEAR ALL ERRORS
3663 016644 005777 162244 TST @RSER ;DID OPI CLEAR IN RSER
3664 016650 001437 BEQ 1$ ;YES
3665 016652 104402 016656 TYPE ,.+2 ;.ASCIZ <15><12>"OPI IN RSER DID NOT CLEAR BY SETTING CL
3666 016746 104040 HLT !DS ;RSDS SHOULD=0
3667 016750 022777 010600 162134 1$: CMP #10600,@RSDS ;DID ERROR BITS CLEAR IN RSDS
3668 ;BY SETTING CLR BIT IN RSCS2
3669 016756 001401 BEQ .+4 ;YES
3670 016760 104040 HLT !DS ;RSDS SHOULD=10600

```

F07

```

3671 ;*****
3672 ;TEST 61 PARITY ERROR TEST
3673 ;*****
3674 016762 104400 TST61: SCOPE
3675
3676 ;MODULES TESTED: M7754, M7770
3677 ;SET "PAT" BIT IN RSCS2. WRITE A DRIVE REGISTER. "PAR" SHOULD SET IN
3678 ;THE DRIVE ERROR REGISTER (RSER) WHICH SHOULD CAUSE "ATA" TO SET IN RSAS
3679 ;AND 'SC' TO SET IN RSCS1.
3680
3681 016764 104414 MRPAR: CLRDK ;CLEAR ALL REGISTERS
3682 016766 042767 000040 162172 BIC #BITS,ONCEE ;CLEAR CLK CNT FLAG
3683 016774 104430 MRIND ;SEND INDEX PULSE TO MR REG
3684 016776 104420 MRCK ;CHECK MAINT TO
3685 017000 022701 22701 ;CHECK MAINT TO
3686 017002 104424 MRINT ;EQUAL 22701
3687 017004 052777 000020 162070 BIS #BIT4,RSCS2 ;INIT MAINT MODE (CLEAR MRSP)
3688 017012 012777 000077 162070 MOV #77,RSDA ;SET THE "PAT" BIT.
3689 ;BY WRITING INTO THIS REGISTER,
3690 017020 022777 000010 162066 CMP #10,RSER ;PAR SHOULD SET IN RSER
3691 017026 001401 BEQ +4 ;DID PAR SET?
3692 017030 104040 HLT !DS ;YES
3693 ;"PAR" IN RSER SHOULD BE SET FOR
3694 ;THE "PAT" BIT WAS SET IN RSCS2
3695 017032 022777 104200 162040 CMP #104200,RSCS1 ;WHEN PROGRAM TRIED TO WRITE INTO RSDA
3696 017040 001401 BEQ +4 ;DID PAR CAUSE SC TO SET?
3697 017042 104044 HLT !DS!DA ;YES
3698 ;SC SHOULD BE SET IN RSCS1 FOR
3699 017044 022777 000077 162036 CMP #77,RSDA ;PAR SHOULD BE SET IN RSER
3700 017052 001401 BEQ +4 ;DID RSDA GET LOADED?
3701 017054 104004 HLT !DA ;YES
3702 ;RSDA SHOULD=77 FOR PAT
3703 ;BIT WAS SET WHEN PROGRAM
3704 017056 104414 CLRDK ;TRIED TO WRITE INTO RSDA
3705 017060 022777 004200 162012 CMP #4200,RSCS1 ;CLEAR ALL ERRORS
3706 017066 001401 BEQ +4 ;DID ERRORS CLEAR?
3707 017070 104044 HLT !DS!DA ;YES
3708 ;SC DID NOT CLEAR BY USING
3709 017072 005777 162016 TST RSER ;THE "CLR" BIT IN RSCS2
3710 017076 001401 BEQ +4 ;DID PAR CLEAR?
3711 017100 104044 HLT !DS!DA ;YES
3712 ;PAR DID NOT CLEAR BY USING
;THE CLR BIT IN RSCS2

```

```

3713 :*****
3714 :TEST 62 MAINTENANCE MODE INTERRUPT TEST
3715 :*****
3716 017102 104400 TST62: SCOPE
3717
3718 :MODULE TESTED M7771
3719 :IN THIS TEST THE INTERRUPT ENABLE BIT IS SET (I.E.).
3720 :A TWO SECTOR WRITE COMMAND IS GIVEN. AN "RMR"
3721 :ERROR IS CREATED WHILE THE FIRST SECTOR IS BEING WRITTEN
3722 :THIS SHOULD CAUSE THE DRIVE TO INTERRUPT AFTER THE FIRST
3723 :SECTOR IS WRITTEN AND THE TRANSFER TO TERMINATE.
3724
3725 017104 012767 000002 162032 MREX: MOV #2,FLAG2 ;CLEAR DRIVE REGISTERS
3726 017112 104414 CLDK ;SETUP FOR INTERRUPT
3727 017114 012737 000200 177776 MOV #200,#PS
3728 017122 012706 000500 MOV #500,SP
3729 017126 052767 000040 162032 BIS #BITS,ONCEE ;SET TYPE CLOCK COUNT FLAG
3730 017134 042767 000600 162024 BIC #600,ONCEE ;CLEAR FLAG BITS
3731 017142 104430 MRIND ;SEND INDEX PULSE TO MR REG
3732 017144 104420 MRCK ;CHECK MR REG
3733 017146 022701 22701 ;TO EQUAL 22701
3734 017150 104424 MRINT ;INIT MAINT MODE (CLEAR MRSP)
3735 ;BY SENDING 2 CLOCK PULSES
3736
3737 :FILL MEMORY DATA BUFFER (INBUF) WITH 128 WORDS (1 SECTOR)
3738 :DATA BUFFER WORDS ARE :A WORD OF ALL 0'S - ALL 1'S
3739 : FLOATING 1'S PATTERN (16 WORDS)
3740 : A PATTERN OF 146314 (110 WORDS)
3741 017152 012702 026610 MOV #INBUF,R2 ;GET LOCATION OF OUTBUF
3742 017156 005022 CLR (R2)+ ;CLEAR 1ST LOCATION
3743 017160 012722 177777 MOV #-1,(R2)+ ;2ND WORD OF ALL ONES
3744 017164 005003 CLR R3 ;CLEAR WORK LOC TO GENERATE
3745 017166 000261 SEC ;A PATTERN OF FLOATING ONES
3746 017170 006103 1$: ROL R3 ;GET PATTERN
3747 017172 103402 BCS 2$ ;DONE GET OUT
3748 017174 010322 MOV R3,(R2)+ ;FILL BUFFER
3749 017176 000774 BR 1$ ;CONT
3750 017200 012703 000156 2$: MOV #110,R3 ;FILL REMAINING PORTION OF
3751 017204 012704 146314 MOV #146314,R4 ;BUFFER WITH A PATTERN OF 146314
3752 017210 010422 3$: MOV R4,(R2)+ ;LOAD BUFFER
3753 017212 005303 DEC R3 ;DONE YET?
3754 017214 001375 BNE 3$ ;NO
3755
3756 :SETUP CONTROLLER TO TRANSFER 256 WORDS OF DATA (2 SECTORS)
3757 017216 012777 020024 161704 MOV #INTMR,#RSVEC ;SETUP INTERRUPT VECTOR
3758 017224 012777 000340 161700 MOV #340,#RSVCPS
3759 017232 012777 026610 161646 MOV #INBUF,#RSBA ;LOAD BUS ADDR REG
3760 017240 012777 177400 161636 MOV #177400,#RSWC ;LOAD WORD COUNT REG
3761 017246 012777 000161 161624 MOV #161,#RSCS1 ;LOAD WRITE COMMAND I/E
3762 017254 104446 GETSP ;CLOCK ROUTINE TO GET SECTOR PULSE
3763 ;TO CLEAR OUT COUNTERS AND REGISTERS
3764 ;THAT OTHERWISE COULD NOT BE CLEARED.
3765 017256 104220 HLT !MR ;COULD NOT SET SECTOR PULSE (0)
3766 017260 104450 SPASS ;CLOCK MR REG SP = 1
    
```



```

3812                                     ;DRIVE SHOULD NOW RECEIVE 1ST WORD TO BE WRITTEN
3813
3814 017410 104422                     MRCLK                     ;CLOCK MR REG
3815 017412 104420                     MRCK                      ;CHECK MR REG
3816 017414 171501                     171501                   ;TO EQUAL 171501
3817 017416 104000                     HLT                       ;MR REG=BAD GOOD=CORRECT ANS
3818 017420 104422                     MRCLX                    ;CLOCK MR REG
3819 017422 104420                     MRCK                      ;MR REG SHOULD
3820 017424 025501                     25501                    ;EQUAL 25501
3821 017426 104000                     HLT                       ;MR REG=BAD GOOD=CORRECT ANS
3822 017430 104422                     MRCLK
3823 017432 104420                     MRCK
3824 017434 175501                     175501
3825 017436 104000                     HLT
3826                                     ;PERFORM NEXT STEP 3 TIMES TO FINISH WRITTING PREAMBLE
3827 017440 012767 000003 161534      MREX3: MOV #3,REPT
3828 017446 104422                     MRCLK                     ;CLOCK MR REG
3829 017450 104420                     MRCK                      ;CHECK MR REG
3830 017452 025501                     25501                    ;TO EQUAL 25501
3831 017454 104000                     HLT                       ;MR=BAD GOOD=CORRECT ANS
3832 017456 104422                     MRCLK                     ;CLOCK MR REG
3833 017460 104420                     MRCK                      ;CHECK MR REG
3834 017462 175501                     175501                   ;TO EQUAL 175501
3835 017464 104000                     HLT                       ;MR REG=BAD GOOD=CORRECT ANS
3836 017466 005367 161510            DEC REPT
3837 017472 001365                     BNE MREX3                 ;DONE YES?
3838                                     ;NO LOOP BACK
3839                                     ;MOVE DATA WORD INTO RS04 SHIFT REGISTER
3840
3841 017474 104422                     MRCLK                     ;CLOCK MR REG
3842 017476 104420                     MRCK                      ;CHECK MR REG
3843 017500 027501                     27501                    ;TO EQUAL 27501
3844 017502 104000                     HLT                       ;MR=BAD GOOD=CORRECT ANS
3845 017504 104422                     MRCLK                     ;CLOCK MR REG
3846 017506 104420                     MRCK                      ;MR REG SHOULD
3847 017510 123501                     123501                   ;EQUAL 123501
3848 017512 104000                     HLT                       ;MR=BAD GOOD=CORRECT ANS
3849
3850                                     ;ENCODE SYNC 1
3851
3852 017514 104422                     MRCLK                     ;CLOCK MR REG
3853 017516 104420                     MRCK                      ;MR REG SHOULD NOW
3854 017520 073501                     73501                    ;EQUAL 73501
3855 017522 104000                     HLT                       ;MR=BAD GOOD=CORRECT ANS
3856 017524 012705 026610            MOV #INBUF,R5             ;GET STARTING ADDR FOR DATA BUFFER
3857 017530 011504                     MOV (R5),R4              ;GET DATA
    
```

J07

MAINDEC-11-DERSO-C RH70-RS04 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 88
 DERSDC.SRC 17-MAY-77 14:16 TST62 MAINTENANCE MODE INTERRUPT TEST

3858	017532	012767	002167	161454	MOV	#1143.,WORK	:DOING A 1 SECTOR TRANSFER 127 WORDS
3859							:18 BITS PER WORD-CLOCK LOOPS
3860							:TAKE CARE OF 2 BITS AT A TIME
3861							:127 TIMES 9 EQUALS 1143 LOOPS
3862							:TO GET THROUGH SECTOR (LAST WORD DONE SEPARATELY).
3863	017540	042767	000200	161420	BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
3864	017546	052767	000100	161412	BIS	#BIT6,ONCEE	:SET 1ST TRANSFER WORD FLAG
3865	017554	104432			15: XBIT		:GET 2 BITS OF DATA
3866	017556	104434			CLKD1		:SEND FIRST CLOCK PULSE
3867							:AND CALCULATE MR REG
3868							:FOR CORRECT DATA (MMDT+MMD8)
3869	017560	104000			HLT		:MR REG NOT CORRECT
3870	017562	104436			CLKD2		:SEND 2ND CLOCK PULSE TO
3871							:COMPLETE TRANSFER OF 2 BITS
3872							:CALCULATE CORRECT ANS FOR
3873							:MR REG (MMDT+MMD8)
3874	017564	104000			HLT		:MR=BAD GOOD=CORRECT ANS
3875	017566	032767	000200	161372	BIT	#BIT7,ONCEE	:ON LAST WORD YET
3876	017574	001015			BNE	25	:YES
3877	017576	032767	000400	161362	BIT	#BIT8,ONCEE	:ON CRC WORD YET?
3878	017604	001043			BNE	35	:YES
3879	017606	005367	161402		DEC	WORK	:DONE WITH 127 WORDS?
3880	017612	001360			BNE	15	:NO
3881							
3882	017614	052767	000200	161344	BIS	#BIT7,ONCEE	:SET LAST WORD FLAG
3883	017622	012767	000012	161364	MOV	#10.,WORK	:SET UP TO TRANSFER LAST WORD
3884	017630	005367	161360		25: DEC	WORK	:DONE YET
3885	017634	001347			BNE	15	
3886							
3887	017636	052767	000400	161322	BIS	#BIT8,ONCEE	:SET TRANSFERRING CRC WORD
3888	017644	042767	000200	161314	BIC	#BIT7,ONCEE	:CLEAR LAST WORD FLAG
3889							
3890							:GENERATE RMR ERROR BY ATTEMPTING TO WRITE RSER
3891							:EXC SHOULD THEN BE ASSERTED
3892							
3893	017652	012777	177777	161234	MOV	#-1,RSER	
3894	017660	004767	003572		JSR	PC,GENCRC	:GENERATE CRC WORD
3895							:AND LEAVE IN "WORK"
3896	017664	012702	026610		MOV	#INBUF,R2	:GO TO END
3897	017670	062702	000400		ADD	#400,R2	:OF DATA BUFFER
3898	017674	016712	161314		MOV	WORK,R2	:LOAD CRC WORD
3899	017700	010205			MOV	R2,R5	:RESET POINTER FOR
3900	017702	162705	000002		SUB	#2,R5	:R5 FOR CRC MD
3901	017706	012767	000012	161300	MOV	#10.,WORK	:SETUP TO XFER CRC
3902	017714	005367	161274		35: DEC	WORK	:DONE YET?
3903	017720	001315			BNE	15	:NO

K07

```

3904 ;EBL SHOULD NOW ASSERT AND CRC BE WRITTEN
3905 017722 104422 MRCLK ;CLOCK MR REG TO STEP THROUGH DEAD BAND AREA
3906 017724 104420 MRCK ;CHECK MR REG
3907 017726 153501 153501 ;TO EQUAL 103501
3908 017730 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
3909
3910 ;LOOP 6 TIMES
3911 017732 012767 000006 161242 ;LOOP 6 TIMES #6, REPT
3912 017740 104422 4S: MRCLK ;CLOCK MR REG
3913 017742 104420 MRCK ;CHECK MR REG
3914 017744 003501 3501 ;TO EQUAL 53501
3915 017746 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3916 017750 104422 MRCLK ;CLOCK MR REG
3917 017752 104420 MRCK ;CHECK MR REG
3918 017754 153501 153501 ;TO EQUAL 103501
3919 017756 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3920 017760 005367 161216 DEC REPT ;DONE LOOPING YET?
3921 017764 001365 BNE 4S ;NO
3922
3923 ;FINISH UP
3924 017766 104422 MRCLK ;CLOCK MR REG
3925 017770 104420 MRCK ;CHECK MR REG
3926 017772 003501 3501 ;TO EQUAL 3501
3927 017774 104000 HLT ;MR REG=BAD GOOD=CORRECT ANS
3928 017776 104422 MRCLK ;CLOCK MR REG
3929 020000 104420 MRCK ;CHECK MR REG
3930 020002 151501 151501 ;TO EQUAL 151501
3931 020004 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3932
3933 ;TRANSFER SHOULD NOW BE COMPLETE
3934 020006 104422 MRCLK ;CLOCK MR REG
3935 020010 104420 MRCK ;CHECK MR REG
3936 020012 002701 2701 ;TO EQUAL 2701
3937 020014 104000 HLT ;MR=BAD GOOD=CORRECT ANS
3938 020016 000240 NOP ;STALL FOR TIME
3939 020020 104050 HLT !WC!DS ;SHOULD NEVER GET HERE
3940 020022 000424 BR INTMR1 ;BECAUSE DRIVE SHOULD HAVE INTERRUPTED.
3941 ;CAUSING JUMP TO INTMR.
3942 ;CHECK FOR ASSERTION OF FTS ATTN L
3943
3944 ;NOW TEST CONTROLLER
3945 020024 022777 144260 161046 INTMR: CMP #144260, @RSCS1 ;IS CS1 CORRECT?
3946 020032 001401 BEQ .+4 ;YES
3947 020034 104014 HLT !DA!WC ;YES
3948 020036 022777 177610 161040 5S: CMP #177610, @RSWC ;IS WC REG CORRECT?
3949 020044 001401 BEQ .+4 ;YES
3950 020046 104010 HLT !WC ;WC SHOULD BE = TO 177610
3951 020050 022777 000004 161036 CMP #4, @RSER ;DID RMR SET IN RSER
3952 020056 001401 BEQ .+4 ;YES
3953 020060 104050 HLT !DS!WC ;RSER SHOULD = 4
3954 020062 022777 000001 161020 CMP #1, @RSDA ;DOES RSDA=1
3955 020070 001401 BEQ .+4 ;YES
3956 020072 104004 HLT !DA ;RSDA SHOULD=1
3957 020074 000240 INTMR1: NOP ;DONE

```

```

3958 ;*****
3959 ;TEST 63 DISK ADDRESS OVERFLOW TEST
3960 ;*****
3961 020076 104400 TST63: SCOPE
3962
3963 ;MODULES TESTED: M7754, M7771, M7770
3964 ;SET UP TO TRANSFER 2 SECTORS TO THE DISK, STARTING AT TRACK 77 SECTOR 77
3965 ;TO CAUSE A DISK ADDRESS OVERFLOW CONDITION. ALSO CHECK LAST BLOCK TRANSFER
3966 ;(LBT) BIT TO SET IN THE RSDS REGISTER.
3967
3968 020100 104414 MRAOE: CLARK ;CLEAR ALL REGISTERS
3969 020102 012706 000500 MOV #500,SP ;SETUP STACK POINTER
3970 020106 104430 MRIND ;SEND INDEX PULSE TO MR REG
3971 020110 104420 MRCK ;CHECK MAINT REG
3972 020112 022701 22701 ;TO EQUAL 22701
3973 020114 104424 MRINT ;INITIALIZE MAINT REG BY SENDING
3974 ; 2 CLOCK PULSES (CLEAR MRSP)
3975 020116 012777 007777 160764 MOV #7777, @RSDA ;SETUP DISK ADDRESS
3976 020124 012777 177400 160752 MOV #400, @RSHC ;SETUP FOR A 2 SECTOR TRANSFER
3977 020132 012777 027410 160746 MOV @OUTBUF, @RSBA ;GET OUTPUT BUFFER
3978
3979 ;SETUP BUFFER WITH ALL ONES
3980 020140 012705 027410 MOV @OUTBUF, R5 ;GET STARTING ADDRESS OF OUTBUF
3981 020144 012767 000400 161030 MOV #400, REPT ;LOAD 2 SECTORS
3982 020152 012725 177777 1S: MOV #1, (R5)+ ;WITH WORDS
3983 020156 005367 161020 DEC REPT ;OF ALL ONES
3984 020162 001373 BNE 1S
3985
3986 020164 012777 000061 160706 MOV #61, @RSCS1 ;LOAD WRITE COMMAND
3987 020172 104430 MRIND ;SET INDEX PULSE
3988
3989 ;SUPPLY CLOCKS TO STEP THROUGH A TRACK
3990
3991 020174 012767 000003 161000 MOV #3, REPT
3992 020202 012704 160000 5S: MOV #57344., R4 ;SETUP FOR FAST CLOCK PULSES 172032 CLOCKS
3993 020206 012702 000011 MOV #11, R2 ;(3 X 57344 = 172032)
3994 020212 012703 000001 MOV #1, R3
3995 020216 010277 160702 2S: MOV R2, @RSMR
3996 020222 010377 160676 MOV R3, @RSMR
3997 020226 005304 DEC R4
3998 020230 001372 BNE 2S
3999 020232 005367 160744 DEC REPT
4000 020236 001361 BNE 5S
4001
4002 ;CAUSE "LBT IN RSDS TO SET
4003
4004 020240 104422 MRCLK ;CLOCK AN 11 AND A 1 INTO RSMR
4005 020242 104426 DSCK ;CHECK MR
4006 020244 012400 12400 ;TO EQUAL 12400
4007 020246 104000 HLT ;LBT SHOULD BE SET IN RSDS

```

M07

```

4008
4009
4010      ; ASSERT MAINTENANCE INDEX PULSE TO RESET DRIVE
4011      ; FOR THE SECOND REVOLUTION
4012      020250  104430      MRIND      ; ASSERT MAINT INDEX PULSE
4013      020252  005067  160716      CLR      MCCNT      ; CLEAR THE CLOCK COUNTER
4014      020255  104420      MRCK      ; CHECK MR REG
4015      020260  002501      2501      ; TO EQUAL 2501. SHOULD STILL BE WRITING
4016      020262  104000      HLT
4017
4018      ; SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS04 RESYNC PERIOD
4019      020264  012767  001000  160710  ; SUPPLY ENOUGH CLOCKS TO STEP THROUGH THE RS04 RESYNC PERIOD
4020      020272  104422      4S:      MOV      #512.,REPT      ; CLOCK COUNT TO STEP THRU RESYNC
4021      020274  104420      MRCLK     ; 2ND REVOLUTION
4022      020276  052501      MRCK      ; CHECK MR
4023      020300  104000      52501    ; TO EQUAL 52501
4024      020302  104422      HLT      ; MR=BAD GOOD=CORRECT ANS
4025      020304  104420      MRCLK     ; CLOCK MR REG
4026      020306  002501      MRCK      ; CHECK MR
4027      020310  104000      2501     ; REG TO
4028      020312  005367  160664      HLT      ; EQUAL 2501
4029      020316  001365      DEC      REPT
4030      BNE      4S      ; LOOP TILL DONE
4031
4032      ; SUPPLY 2 CLOCKS TO CAUSE THE SECTOR PULSE TO APPEAR IN
4033      ; THE MR REGISTER AND THE "AOE" ERROR TO APPEAR IN
4034      ; THE RSER REGISTER
4035      020320  104422      AOECK:   MRCLK
4036      020322  104422      MRCLK
4037      020324  104420      MRCK
4038      020326  022301      22301
4039      020330  104000      HLT
4040      020332  022777  001000  160554      CMP      #1000,RSER
4041      020340  001401      BEQ      1S
4042      020342  104040      HLT      !DS
4043      020344  022777  152600  160540  1S:      CMP      #152600,RSDS
4044      020352  001401      BEQ      2S
4045      020354  104040      HLT      !DS
4046
4047      020356  104414      2S:      CLRDK
4048      020360  005777  160530      TST      RSER
4049      020364  001401      BEQ      3S
4050      020366  104040      HLT      !DS
4051      020370  022777  010600  160514  3S:      CMP      #10600,RSDS
4052      020376  001401      BEQ      +4
4053      020400  104040      HLT      !DS
4054
; CAUSE SECTOR PULSE AND AOE ERROR
; CHECK FOR SECTOR PULSE
; IN RSMR
; MR=BAD GOOD=CORRECT ANS
; DID AOE SET IN RSER?
; AOE SHOULD BE SET IN RSER
; RSER SHOULD EQUAL 1000
; IS RSDS CORRECT
; YES
; 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

```

N07

4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096

020402 104400

020404 032767 004000 157156
020412 001002
020414 000137 021130
020420 005067 160520
020424 104414
020426 012767 177777 160574
020434 005367 160570
020440 001375
020442 042767 000040 160516
020450 012777 160000 160426
020456 012767 177777 006124
020464 052777 000010 160410
020472 012777 026610 160406
020500 012767 177777 160474
020506 012777 000061 160364
020514 105777 160360
020520 100404
020522 005367 160454
020526 001372
020530 104000
020532 005777 160342
020536 100002
020540 104050
020542 000433

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

;TEST 64 MAINTENANCE MODE VERIFY TEST

TST64: SCOPE

;MODULE TESTED G182

MRVR: BIT #BIT11,SWR ;DO THIS TEST?
BNE 3S ;YES
JMP @#INFTST ;NO
3S: CLR @FLAG2 ;SET VERIFY TEST FLAG
CLRK ;CLEAR ALL DRIVES
MOV #177777,WORKS ;STALL TO
4S: DEC WORKS ;RESYNC DRIVE
BNE 4S ;TIMING LOGIC
BIC #BITS,ONCEE ;CLEAR CLK CNT
MOV #-20000,@RSWC ;WRITE ONE TRACK - 8K WDS
MOV #177777,INBUF ;WRITE A PATTERN 12525
BIS #BIT3,@RSCS2 ;SET BAI BIT
MOV #INBUF,@RSBA ;SET DATA MD
MOV #177777,REPT ;SETUP WAIT LOOP
MOV #61,@RSCS1 ;GO WRITE
1S: TSTB @RSCS1 ;DONE YET?
BMI 2S ;YES
DEC REPT ;DECREMENT COUNTER WAITING
BNE 1S ;FOR READY
HLT ;READY NEVER CAME UP
2S: TST @RSCS1 ;ANY ERRORS?
BPL MRVR1 ;NO
HLT !DS!WC ;STOP HERE TILL THIS PROBLEM IS FIXED TRY DZRSB DIAG
BR TBDIA ;TYPE MESSAGE

4097	020544	104414			MRVR1:	CLDK			:CLEAR ALL REGISTERS
4098	020546	012777	160000	160330		MOV	#-20000, ARSMC		:SETUP WC
4099	020554	052777	000010	160320		BIS	#BIT3, ARSCS2		:SET BAI
4100	020562	012777	026610	160316		MOV	#INBUF, ARSBA		:SETUP RSBA
4101	020570	012767	177777	160404		MOV	#177777, REPT		:SETUP WAIT LOOP
4102	020576	012777	000051	160274		MOV	#51, ARSCS1		:DO A WRITE CHECK TO VERIFY DISK
4103	020604	105777	160270		1S:	TSTB	ARSCS1		:TEST
4104	020610	100404				BMI	2S		:FOR READY TO COME BACK
4105	020612	005367	160364			DEC	REPT		:WAIT
4106	020616	001372				BNE	1S		
4107	020620	104000				HLT			:READY NEVER CAME BACK
4108	020622	005777	160252		2S:	TST	ARSCS1		:ANY ERRORS?
4109	020626	100032				BPL	MRVRR		:NO
4110	020630	104050				HLT	!DS!WC		:STOP HERE WC FAILED
4111									:GO TO DZRSB DIAG
4112									:BEFORE TRYING TO DEBUG THIS TEST
4113	020632				TBDIA:				
4114	020632	104402	020636			TYPE	.+2		:ASCIZ <15><12>"FAILED VERIFY TEST --- RUN DZRSB DIAGNO
4115	020714	000137	011456		MRVRR:	JMP	#MRVRT		:GO WRITE IN MAINTENANCE MODE
4116									:NOW CHECK TO SEE IF DRIVE WAS WRITTEN ON IN MAINTENANCE MODE
4117									
4118	020720	104414			MRVR2:	CLDK			:CLEAR ALL REGISTERS
4119	020722	012767	177777	160264		MOV	#177777, WORK		:STALL - TO RESPONSE
4120	020730	005367	160260		3S:	DEC	WORK		:INDEX PULSE
4121	020734	001375				BNE	3S		:ON DRIVE
4122	020736	012777	160000	160140		MOV	#-20000, ARSMC		:SETUP WC FOR 1 TRACK
4123	020744	052777	000010	160130		BIS	#BAI, ARSCS2		:SET BAI
4124	020752	012777	026610	160126		MOV	#INBUF, ARSBA		:SETUP RSBA
4125	020760	012767	177777	005622		MOV	#177777, INBUF		:SETUP FOR COMPARE
4126	020766	012777	000051	160104		MOV	#51, ARSCS1		:DO A WRITE CHECK
4127	020774	105777	160100		1S:	TSTB	ARSCS1		:TEST FOR
4128	021000	100375				BPL	1S		:READY TO COME BACK
4129	021002	032777	040000	160072		BIT	#MCE, ARSCS2		:DID MCE SET?
4130	021010	001442				BEQ	2S		:NO
4131	021012	104402	021016			TYPE	.+2		:ASCIZ <15><12> "WRITE AMPLIFIER DID NOT GET DISABLED B
4132	021112	104040				HLT	!DS		
4133	021114	000404				BR	4S		:GET OUT
4134	021116	005777	157756		2S:	TST	ARSCS1		:ANY ERRORS?
4135	021122	100001				BPL	4S		:NO
4136	021124	104040				HLT	!DS		:SHOULD NOT BE ANY ERRORS
4137									:TRY THE DZRSB DIAGNOSTIC
4138	021126	000240			4S:	NOP			
4139									
4140	021130	000137	002116		INFTST:	JMP	#TRYNX		:GET NEXT DRIVE

```

4141          .SBTTL          SDONE - BELL AND SCOPE ROUTINE
4142
4143 021134 104400          DONE:  SCOPE          ; TERMINATING SCOPE FOR LOOPING
4144 021136 062767 000001 157642      ADD      #1,PCNT+2      ; ADD 1 TO THE PASS COUNT
4145 021144 005567 157634          ADC      PCNT          ; MAKE IT DOUBLE PREC.
4146 021150 032737 002000 177570      BIT      #SW10,@SWR    ; RING THE BELL?
4147 021156 001004          BNE      4S           ; NO!
4148 021160 104402 021164          TYPE     .+2          ; .ASCIZ <BELL><177>
4149 021170 013700 000042          4S:     MOV      @#42,RO ; GET MONITOR ADDRESS
4150 021174 001405          BEQ     SEND1        ; IF NONE
4151 021176 000005          RESET
4152 021200 004710          SENDAD: JSR     7,(0)   ; GO TO MONITOR
4153 021202 000240 000240 000240      240,240,240          ; SAVE ROOM FOR ACT11
4154 021210 000167 000002          SEND1: JMP     MULSYS ; RETURN
4155
4156 021214 000000          .TBIT:  0           ; T BIT FLAG
4157
4158          ;MULTI DRIVE SYSTEM?
4159
4160          MULSYS:
4161 021216 104402 021222          TYPE     .+2          ; .ASCIZ <15><12>"END OF PASS"
4162 021240 005067 157544          CLR     LAD          ;
4163 021244 005067 157530          CLR     ICNT        ;
4164 021250 032767 000020 157710      BIT     #BIT4,ONCEE ; MULTI DRVIE?
4165 021256 001002          BNE     1S           ; NO
4166 021260 000137 001672          JMP     @#MULTII    ; YES
4167 021264 000137 002322          1S:     JMP     @#NOWGO ; TEST ONLY ONE DRIVE
4168
4169          ;ERROR TYPEOUT ROUTINE FOR NO-OP TEST
4170
4171 021270 032767 000004 157670      NOPERR: BIT     #BIT2,ONCEE ; WERE WE HERE BEFORE?
4172 021276 001031          BNE     1S           ; YES
4173 021300 052767 000004 157660      BIS     #BIT2,ONCEE ; SET BEEN HERE BEFORE FLAG
4174 021306 104402 021312          TYPE     .+2          ; .ASCIZ <15><12>"ERROR CAUSED BY NO-OP FUNCTION "
4175 021354 016746 157634          MOV     WORD, -(6)   ; PUT WORK ON STACK
4176 021360 104406          TYPES
4177 021362 000207          1S:     RTS      PC   ; TYPE STACK IN OCTAL - SUPRESS

```

4178	021364	104402	021400		CHG:	TYPE	REGCHG		:TYPE MESSAGE
4179	021370	016746	157620			MOV	WORK,-(6)		:PUT WORK ON STACK
4180	021374	104406				TYPES			:TYPE STACK IN OCTAL - SUPRESS
4181	021376	000207				RTS	PC		
4182									
4183	021400	044103	047101	042507	REGCHG:	.ASCIZ	"CHANGED WITH NO-OP FUNCTION "		
4184	021406	020104	044527	044124					
4185	021414	047040	026517	050117					
4186	021422	043040	047125	052103					
4187	021430	047511	020116	000					
4188									
4189	021435	015	051012	051115	TRMR:	.ASCIZ	<15><12>"RMR DID NOT SET BY WRITING INTO "		
4190	021442	020040	044504	020104					
4191	021450	047516	020124	042523					
4192	021456	020124	054502	053440					
4193	021464	044522	044524	043516					
4194	021472	044440	052116	020117					
4195	021500	000							
4196		021502							
4197						.EVEN			
4198	021502	104422			.MRINT:	MRCLK			:CLOCK THE MAINT REG WITH AN 11 AND A 1
4199	021504	104422				MRCLK			:SAME
4200	021506	000002				RTI			:RETURN
4201									
4202	021510	012777	000011	157406	.MRCLK:	MOV	#11,RSMR		:CLOCK THE
4203	021516	012777	000001	157400		MOV	#1,RSMR		:MAINT REG
4204	021524	062767	000001	157444		ADD	#1,MCCNT+2		:ADD 1 TO CLOCK COUNT
4205	021532	005567	157436			ADC	MCCNT		:MAKE DOUBLE PRECISION
4206	021536	000002				RTI			
4207									
4208	021540	017700	157360		.MRCK:	MOV	RSR,BAD		:GET THE CONTENTS OF RSMR
4209	021544	017601	000000			MOV	2(SP),GOOD		:GET THE CORRECT ANSWER
4210	021550	062716	000002			ADD	#2,(SP)		:UPDATE THE RETURN ADDRESS FOR AN ERROR
4211	021554	020100				CMP	GOOD,BAD		:IS THE MR REG CORRECT?
4212	021556	001002				BNE	1\$:NO EXIT
4213	021560	062716	000002			ADD	#2,(SP)		:UPDATE RETURN ADDRESS TO SKIP THE HLT FOR CORRECT ANS
4214	021564	000002			1\$:	RTI			:RETURN
4215									
4216									
4217	021566	012777	000021	157330					:SEND INDEX PULSE TO THE MAINTENANCE REGISTER
4218	021574	012777	000001	157322	.MRIND:	MOV	#21,RSR		:SEND INDEX
4219	021602	000002				MOV	#1,RSR		:PULSE TO MR REG
4220	021604	017700	157302			RTI			
4221	021610	017601	000000		.DSCK:	MOV	RSR,BAD		:GET THE CONTENTS OF RSDS
4222	021614	062716	000002			MOV	2(SP),GOOD		:GET THE CORRECT ANS
4223	021620	020100				ADD	#2,(SP)		:UPDATE THE RETURN ADDR FOR AN ERROR
4224	021622	001002				CMP	GOOD,BAD		:IS RSDS CORRECT
4225	021624	062716	000002			BNE	1\$:NO EXIT
4226	021630	000002			1\$:	ADD	#2,(SP)		:UPDATE RETURN ADDR TO SKIP THE HLT FOR CORRECT ANS
						RTI			

E08

```

4227
4228
4229
4230
4231
4232
4233
4234 021632 032767 000100 157326 .XBIT: BIT      #BIT6,ONCEE  ;1ST 2 BITS OF 1ST WORD?
4235 021640 001427          157276 BEQ      25      ;NO
4236 021642 012767 000001 157276 MOV      #1,LSTEV ;SET LAST EVEN BIT TRANSFERRED TO A 1
4237 021650 012767 000001 157272 MOV      #1,LSTOD ;SET LAST ODD BIT TRANSFERRED TO A 1
4238                                     ;THIS SETS UP THE SYNC 1 BITS AT END OF PREAMBLE
4239                                     ;FOR THE TOP AND BOTTOM
4240                                     ;BITS IN THE MR REGISTER
4241 021656 042767 000100 157302          BIC      #BIT6,ONCEE ;CLEAR 1ST WORD TRANSFER FLAG
4242 021664 005067 157316          CLR      CLKCNT      ;CLEAR CLOCK COUNTER AT START OF EACH WORD
4243 021670 032767 000400 157270 4S:  BIT      #BIT8,ONCEE ;CRC WORD BEING WRITTEN?
4244 021676 001042          BNE      15      ;YES
4245 021700 005067 157250          CLR      NOWOD      ;NO, LOAD EVEN
4246 021704 005067 157242          CLR      NOWEV      ;AND ODD WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD.
4247 021710 012767 000010 157306 6S:  MOV      #8.,WORK3 ;8 LOOPS FOR REMAINING 16 BITS OF WORD
4248 021716 000002          RTI
4249 021720 016767 157230 157222 2S:  MOV      NOWOD,LSTOD
4250 021726 016767 157220 157212 MOV      NOWEV,LSTEV ;SAVE LAST 2 BITS TRANSFERRED
4251 021734 005767 157264          TST      WORK3      ;DONE WITH WORD YET?
4252 021740 001004          BNE      35      ;NO
4253 021742 062705 000002          ADD      #2,R5      ;UPDATE BUFFER WD
4254 021746 011504          MOV      (R5),R4    ;GET DATA WD
4255 021750 000745          BR       45      ;GET BITS 16 & 17
4256 021752 005067 157176          CLR      NOWOD      ;CLEAR PRESENT ODD BIT
4257 021756 006104          ROL      R4         ;GET NEXT ODD DATA BIT
4258 021760 006167 157170          ROL      NOWOD      ;SAVE IT IN ODD BIT
4259 021764 005067 157162          CLR      NOWEV      ;CLEAR PRESENT EVEN BIT
4260 021770 006104          ROL      R4         ;GET NEXT EVEN BIT
4261 021772 006167 157154          ROL      NOWEV      ;SAVE IT IN EVEN BIT
4262 021776 005367 157222          DEC      WORK3      ;KEEP COUNT OF BITS IN THE WORD
4263 022002 000002          RTI                ;RETURN
4264
4265
4266 022004 005067 157144          1S:  CLR      NOWOD      ;CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS. 0 & 1 ARE ALWAYS 0
4267 022010 006104          ROL      R4         ;GET BITS 17
4268 022012 006167 157136          ROL      NOWOD      ;AND 16
4269 022016 005067 157130          CLR      NOWEV      ;FOR CRC WORD
4270 022022 006104          ROL      R4
4271 022024 006167 157122          ROL      NOWEV
4272 022030 000727          BR       65      ;CONTINUE
  
```


F08

MAINDEC-11-DERSO-C RH70-RSD4 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 97
 DERSOC.SRC 17-MAY-77 14:16 SDONE - BELL AND SCOPE ROUTINE

```

4273 ;CLOCK ROUTINE (1ST OF TWO) WHICH IS USED TO CLOCK TWO BITS OF
4274 ;DATA TO THE DRIVE AT A TIME. THIS ROUTINE ALSO CHECKS THE PREVIOUS
4275 ;BITS THAT HAVE BEEN TRANSFERRED AND CALCULATES WHICH STATE
4276 ;THE MMDT BIT (BIT 14 IN THE MR REG) AND MMD8 BIT (BIT 12 IN THE MR REG) SHOULD BE IN
4277
4278
4279 022032 104422 .CLKD1: MRCLK ;CLOCK MR REG WITH AN 11 AND A 1
4280 022034 005003 CLR R3 ;CLEAR WORK LOCATION
4281 022036 005767 157112 TST NOWOD ;TEST ODD BIT NOW BEING SENT FOR A 1 OR A 0
4282 022042 001005 BNE TSTEV8 ;NOW TEST EVEN DATA BIT ON 1ST CLOCK
4283
4284 022044 005767 157100 1S: TST LSTOD ;NOW BIT IS A 1 MMD8 IS 0
4285 022050 001002 BNE TSTEV8 ;TEST THE LAST ODD DATA BIT THAT WAS SENT
4286
4287
4288 022052 052703 010000 2S: BIS #BIT12,R3 ;SET MMD8 FOR LATER COMPARE WITH MR REG
4289
4290 ;NOW TEST FOR EVEN BITS BEING SENT
4291
4292 022056 005767 157070 TSTEV8: TST NOWEV ;TEST EVEN BIT NOW BEING TRANSFERRED
4293
4294 022062 001005 BNE 1S ;FOR EITHER A 1 OR A 0
4295 022064 005767 157056 TST LSTEV ;NOW BIT IS A 1
4296 022070 001002 BNE 1S ;WAS LAST EVEN DATA BIT A 0?
4297 022072 052703 040000 BIS #BIT14,R3 ;NO LAST EVEN DATA BIT WAS A 1
4298 022076 012701 123501 1S: MOV #123501,GOOD ;MMDT SHOULD BE SET
4299 022102 050301 BIS R3,GOOD ;GET CORRECT ANS
4300 022104 004767 001200 JSR PC,MRCAL ;FOR MR REG
4301 022110 017700 157010 MOV #RSMR,BAD ;DETERMINE STATE OF SB & LSR BITS
4302 022114 020100 CMP GOOD,BAD ;GET CONTENTS OF MR REG
4303 022116 001002 BNE 2S ;IS MR REG CORRECT?
4304 022120 062716 000002 ADD #2,(SP) ;NO TYPE OUT MR REG
4305 022124 000002 2S: RTI ;UPDATE RETURN ADDR FOR CORRECT ANS
;RETURN

```

G08

```

4306 ;SECOND CLOCK ROUTINE WHICH WILL FINISH TRANSFERRING THE TWO DATA BITS
4307 ;THIS ROUTINE WILL CALCULATE WHAT MMDT AND MMDB SHOULD EQUAL IN THE
4308 ;MAINTENANCE REGISTER
4309
4310 .CLKD2: MRCLK ;CLOCK MR REG
4311 TST NOWOD ;IS THE PRESENT DATA BIT A 1?
4312 BEQ 1$ ;NO IT IS A 0
4313 BIS #BIT12,R3 ;SET MMDB FOR BIT BEING SENT IS A 1
4314 BR 2$
4315 BIC #BIT12,R3 ;CLEAR MMDB FOR PRESENT BIT IS A 0
4316 TST NOWEV ;IS PRESENT EVEN BIT A 1
4317 BEQ 3$ ;NO IT IS A 0
4318 BIS #BIT14,R3 ;IT IS A 1 SET MMDT
4319 BR 4$
4320 BIC #BIT14,R3 ;PRESENT BIT IS A 0 CLEAR MMDT
4321 MOV #23501,GOOD ;GET CORRECT ANS
4322 BIS R3,GOOD ;FOR MR REG
4323 JSR PC,MRCAL ;DETERMINE STATE OF SB & LSR BITS
4324 MOV #RSMR,BAD ;GET CONTENTS OF MR REG
4325 CMP GOOD,BAD ;IS MR REG CORRECT?
4326 BNE 5$ ;NO TIMEOUT ERROR
4327 ADD #2,(SP) ;UPDATE RETURN ADDR FOR CORRECT ANS
4328 RTI ;RETURN
4329
4330 ;TIMEOUT ROUTINE TO DETERMINE WHICH IC FAILED IN CRC TEST2
4331 ;AND TO TYPE IT OUT
4332
4333 CRCTYP: MOV #CRCTAB,WORK ;GET STARTING LOC OF IC TABLE
4334 MOV #1,WORK1 ;SETUP TO TEST FIRST CHIP
4335 BIT WORK1,SAVEE ;WAS IT THIS BIT?
4336 BNE 2$ ;YES TYPE IT
4337 ADD #6,WORK ;NO INDEX TABLE POINTER
4338 ROL WORK1 ;SETUP TO TEST NEXT CHIP
4339 BR 1$ ;NOW TEST IT
4340 JSR PC,@WORK ;TYPE OUT CHIP
4341 TYPE #2 ;".ASCIZ " IN THE CRC REG SHOULD BE SET"
4342 RTS PC
  
```

;TABLE FOR CRC TEST 2 TYPEOUT ROUTINE

4343				CRCTAB: TYPE	E302
4344				RTS	PC
4345	022330	104402	022470	TYPE	E305
4346	022334	000207		RTS	PC
4347	022336	104402	022476	TYPE	E307
4348	022342	000207		RTS	PC
4349	022344	104402	022504	TYPE	E3010
4350	022350	000207		RTS	PC
4351	022352	104402	022512	TYPE	E3012
4352	022356	000207		RTS	PC
4353	022360	104402	022521	TYPE	E3015
4354	022364	000207		RTS	PC
4355	022366	104402	022530	TYPE	E242
4356	022372	000207		RTS	PC
4357	022374	104402	022537	TYPE	E245
4358	022400	000207		RTS	PC
4359	022402	104402	022545	TYPE	E247
4360	022406	000207		RTS	PC
4361	022410	104402	022553	TYPE	E2410
4362	022414	000207		RTS	PC
4363	022416	104402	022561	TYPE	E2412
4364	022422	000207		RTS	PC
4365	022424	104402	022570	TYPE	E2415
4366	022430	000207		RTS	PC
4367	022432	104402	022577	TYPE	E192
4368	022436	000207		RTS	PC
4369	022440	104402	022606	TYPE	E197
4370	022444	000207		RTS	PC
4371	022446	104402	022614	TYPE	E1910
4372	022452	000207		RTS	PC
4373	022454	104402	022622	TYPE	E1915
4374	022460	000207		RTS	PC
4375	022462	104402	022631	TYPE	
4376	022466	000207		RTS	PC

4377	022470	031505	026460	000062	E302:	.ASCIZ	"E30-2"
4378	022476	031505	026460	000065	E305:	.ASCIZ	"E30-5"
4379	022504	031505	026460	000067	E307:	.ASCIZ	"E30-7"
4380	022512	031505	026460	030061	E3010:	.ASCIZ	"E30-10"
4381	022520	000					
4382	022521	105	030063	030455	E3012:	.ASCIZ	"E30-12"
4383	022526	000062					
4384	022530	031505	026460	032461	E3015:	.ASCIZ	"E30-15"
4385	022536	000					
4386	022537	105	032062	031055	E242:	.ASCIZ	"E24-2"
4387	022544	000					
4388	022545	105	032062	032455	E245:	.ASCIZ	"E24-5"
4389	022552	000					
4390	022553	105	032062	033455	E247:	.ASCIZ	"E24-7"
4391	022560	000					
4392	022561	105	032062	030455	E2410:	.ASCIZ	"E24-10"
4393	022566	000060					
4394	022570	031105	026464	031061	E2412:	.ASCIZ	"E24-12"
4395	022576	000					
4396	022577	105	032062	030455	E2415:	.ASCIZ	"E24-15"
4397	022604	000065					
4398	022606	030505	026471	000062	E192:	.ASCIZ	"E19-2"
4399	022614	030505	026471	000067	E197:	.ASCIZ	"E19-7"
4400	022622	030505	026471	030061	E1910:	.ASCIZ	"E19-10"
4401	022630	000					
4402	022631	105	034461	030455	E1915:	.ASCIZ	"E19-15"
4403	022636	000065					

JOB

M3INDEC-11-DERSD-C
 DERSDC.SRC

17-MAY-77 14:16

RH70-RS04 MAINTENANCE MODE DIAGNOSTIC
 SDOE - BELL AND SCOPE ROUTINE

MACY11 27(1006)

17-MAY-77 15:36 PAGE 101

```

4404                               :GET TWO BITS OF DATA FROM INBUF
4405                               :FOR READING FROM DRIVE TO DETERMINE THE
4406                               :STATE OF MR0T AND MR0B IN THE MR REG.
4407
4408 022640 005767 156360          .RBIT: TST      WORK3      :STARTING NEW WD?
4409 022644 001031                BNE      3$          :NO
4410 022646 062705 000002          ADD      #2,R5      :UPDATE BUFFER WD
4411 022652 011504                MOV      (R5),R4    :GET DATA WD
4412 022654 005067 156326          CLR      CLKCNT     :CLEAR CLOCK COUNTER AT START OF EACH WD
4413 022660 032767 000400 156300  BIT      #BITB,ONCE :ON CRC WD?
4414 022666 001035                BNE      1$          :YES
4415 022670 032767 000040 156246  BIT      #BITS,FLAG2 :IN CRC TEST ???
4416 022676 001404                BEQ      7$          :NO
4417 022700 012767 000010 156316  MOV      #8.,WORK3
4418 022706 000410                BR       3$
4419 022710 005067 156240          7$: CLR      NOWOD     :LOAD EVEN & ODD BITS
4420 022714 005067 156232          CLR      NOWEV     :WITH 0 FOR BITS 16 & 17 IN RS04 DATA WORD
4421 022720 012767 000010 156276  6$: MOV      #8.,WORK3 :8 LOOPS FOR REMAINING 16 BITS OF WORD
4422 022726 000002                RTI
4423 022730 005067 156220          3$: CLR      NOWOD     :CLEAR PRESENT ODD BIT
4424 022734 006104                ROL      R4         :GET NEXT ODD DATA BIT
4425 022736 006167 156212          ROL      NOWOD     :SAVE IT IN ODD BIT
4426 022742 005067 156204          CLR      NOWEV     :CLEAR PRESENT EVEN BIT
4427 022746 006104                ROL      R4         :GET NEXT EVEN BIT
4428 022750 006167 156176          ROL      NOWEV     :SAVE IT IN EVEN BIT
4429 022754 005367 156244          DEC      WORK3     :KEEP COUNT OF BITS IN THE WORD
4430 022760 000002                RTI               :RETURN
4431                               :CRC WORD IS BEING WRITTEN BIT 17 & 16 ARE DATA BITS, 0 & 1 ARE ALWAYS 0
4432 022762 005067 156166          1$: CLR      NOWOD     :GET BITS 17
4433 022766 006104                ROL      R4         :AND 16
4434 022770 006167 156160          ROL      NOWOD     :FOR CRC WORD
4435 022774 005067 156152          CLR      NOWEV
4436 023000 006104                ROL      R4
4437 023002 006167 156144          ROL      NOWEV
4438 023006 000744                BR       6$          :CONTINUE

```

K08

4439	023010	004767	000236	.CLKR1:	JSR	PC, CALRTB	; CALCULATE TOP AND BOTTOM BITS FOR MR REG
4440	023014	012703	000011		MOV	#11, R3	; SETUP CLOCK BITS
4441	023020	056703	156170	CLOCK:	BIS	WORK, R3	; SET TOP & BOTTOM BITS
4442	023024	010377	156074		MOV	R3, RSMR	; SEND
4443	023030	042703	000010		BIC	#BIT3, R3	; CLOCK
4444	023034	010377	156064		MOV	R3, RSMR	; PULSE
4445	023040	062767	000001	156130	ADD	#1, MCCNT+2	; INCREMENT
4446	023046	005567	156122		ADC	MCCNT	; CLOCK COUNT
4447	023052	012701	023601		MOV	#23601, GOOD	; CALCULATE CORRECT ANS FOR MR REG
4448	023056	032767	000004	156060	BIT	#BIT2, FLAG2	; WRITE CK TEST?
4449	023064	001402			BEQ	7\$; NO
4450	023066	052701	000100		BIS	#BIT6, GOOD	; YES SET RD IN MR REG
4451	023072	050301		7\$:	BIS	R3, GOOD	
4452	023074	042701	000010		BIC	#BIT3, GOOD	; CLEAR MCLK
4453	023100	032767	000400	156060	BIT	#BIT8, ONCEE	; ON CRC MD?
4454	023106	001406			BEQ	5\$; NO
4455	023110	022767	000011	156064	CMP	#11, REPT	; SHOULD CRCW BE SET?
4456	023116	001402			BEQ	5\$; YES
4457	023120	042701	020000		BIC	#20000, GOOD	; CLEAR CRCW
4458	023124	032767	000001	156012	5\$:	BIT	#BIT0, FLAG2
4459	023132	001004			BNE	1\$; SHOULD SDCLK BE SET?
4460	023134	052767	000001	156002	BIS	#BIT0, FLAG2	; YES
4461	023142	000405			BR	2\$; NO
4462	023144	052701	100000	1\$:	BIS	#BIT15, GOOD	; CONTINUE
4463	023150	042767	000001	155766	BIC	#BIT0, FLAG2	; SET IT
4464	023156	005367	156022	2\$:	DEC	REPT1	; CLEAR FLAG FOR SDCLK FOR NEXT CLOCK PULSE
4465	023162	001017			BNE	6\$; SHOULD SB SET?
4466	023164	012767	000022	156012	MOV	#18, REPT1	; NO
4467	023172	052701	004000		BIS	#BIT11, GOOD	; RESET SB COUNTER
4468	023176	032767	000400	155762	3\$:	BIT	#BIT8, ONCEE
4469	023204	001406			BEQ	6\$; SET SB
4470	023206	022767	000022	155770	CMP	#22, REPT1	; ON CRC MD?
4471	023214	001002			BNE	6\$; NO
4472	023216	052701	020000		BIS	#20000, GOOD	; SHOULD SB AND CRCW BE SET ?
4473	023222	017700	155676	6\$:	MOV	RSMR, BAD	; NO
4474	023226	020100			CMP	GOOD, BAD	; SET SB AND CRCW
4475	023230	001002			BNE	4\$; GET MR REG
4476	023232	062716	000002		ADD	#2, (SP)	; IS RSMR CORRECT?
4477	023236	000002		4\$:	RTI		; NO
4478							; YES
4479	023240	004767	000006	.CLKR2:	JSR	PC, CALRTB	; RETURN
4480	023244	012703	050011		MOV	#50011, R3	
4481	023250	000663			BR	CLOCK	

```

4482 ;CALCULATE THE STATE OF MRDT AND MRDB FROM CURRENT INPUT BITS
4483 ;LOCATION WORK CONTAINS CORRECT DATA FOR MRDT AND MRDB
4484 023252 005067 155736 CALRTB: CLR WORK ;CLEAR WORK LOCATION
4485 023256 005767 155672 TST NOWOD ;IS CURRENT ODD BIT A 0?
4486 023262 001403 BEQ 1$ ;YES
4487 023264 052767 000004 155722 BIS #BIT2,WORK ;NO SET MRDB
4488 023272 005767 155654 1$: TST NOWEV ;IS CURRENT EVEN BIT A 0?
4489 023276 001403 BEQ 2$ ;YES
4490 023300 052767 000040 155706 BIS #BITS,WORK ;NO SET MRDT
4491 023306 000207 2$: RTS PC ;RETURN
4492
4493 ;CALCULATE MR REG TO DETERMINE THE STATE OF THE CRC-SB AND LSR BITS
4494 ;ON THE DIFFERENT CLOCKS ON THE DIFFERENT WORDS THROUGHOUT THE SECTOR
4495
4496 023310 005267 155672 MRCAL: INC CLKCNT ;ADD ONE TO CLOCK COUNT OF WORD
4497 023314 032767 000200 155644 BIT #BIT7,ONCEE ;TRANSFERRING LAST WORD?
4498 023322 001026 BNE LSTWD ;YES
4499 023324 032767 000400 155634 BIT #BIT8,ONCEE ;TRANSFERRING CRC WORD?
4500 023332 001040 BNE CRCWD ;YES
4501 023334 022767 000010 155644 CMP #8.,CLKCNT ;CLOCK COUNT 8 OR GREATER?
4502 023342 101401 BLOS 1$ ;YES
4503 023344 000414 BR 2$ ;GET OUT
4504 023346 022767 000021 155632 1$: CMP #17.,CLKCNT ;CLOCK COUNT 17 OR GREATER?
4505 023354 101410 BLOS 2$ ;YES GET OUT
4506
4507 023356 052701 004000 BIS #BIT11,GOOD ;SET SB BIT
4508 023362 022767 000017 155616 CMP #15.,CLKCNT ;SHOULD LSR BE CLEARED
4509 023370 001002 BNE 2$ ;NO
4510 023372 042701 002000 BIC #BIT10,GOOD ;CLEAR LSR
4511 023376 000207 2$: RTS PC ;RETURN
4512
4513 ;CALCULATE MR FOR LAST DATA WORD
4514 023400 022767 000016 155600 LSTWD: CMP #14.,CLKCNT ;IS THIS CLOCK 14 OR LESS?
4515 023406 103011 BHS 2$ ;YES GETOUT
4516 023410 022767 000017 155570 CMP #15.,CLKCNT ;IS THIS CLOCK 15?
4517 023416 001003 BNE 1$ ;NO
4518 023420 042701 002000 BIC #BIT10,GOOD ;YES CLEAR LSR
4519 023424 000402 BR 2$ ;GET OUT
4520 023426 042701 020000 1$: BIC #BIT13,GOOD ;CLEAR CRCW BIT
4521 023432 000207 2$: RTS PC
4522
4523 ;CALCULATE MR FOR CRC WORD
4524
4525 023434 042701 020000 CRCWD: BIC #BIT13,GOOD ;CLEAR CRCW BIT
4526 023440 022767 000017 155540 CMP #17,CLKCNT ;IS THIS CLOCK 17?
4527 023446 001002 BNE 1$ ;NO
4528 023450 042701 002000 BIC #BIT10,GOOD ;CLEAR LSR BIT
4529 023454 000207 1$: RTS PC ;RETURN
    
```

M08

```

4530
4531      ;GENERATE A CRC WORD FROM THE DATA BUFFER
4532      ;AND LEAVE THE CRC WORD IN "WORK" LOCATION
4533      ;EXIT ROUTINE WITH RTS PC
4534
4535 023456 012767 000200 155516 GENCRC: MOV      #128.,REPT      ;128 WORDS PER SECTOR
4536 023464 032767 000040 155452      BIT      #BITS,FLAG2    ;IN CRC TEST?
4537 023472 001403          BEQ      13$,          ;NO
4538 023474 012767 000220 155500      MOV      #144.,REPT    ;YES
4539 023502 012705 026610          13$:  MOV      #INBUF,R5    ;GET STARTING ADDR OF OUTPUT BUFFER
4540 023506 011504          MOV      (R5),R4      ;GET DATA WD
4541 023510 005067 155502          CLR      WORK0       ;CLEAR WORK LOCATION
4542
4543      ;INBIT CONTAINS PRESENT INPUT BIT
4544      ;WK15 = BIT15 OF CRC AT TIME T
4545      ;WORK0 = CRC AT TIME T + DURING FINAL MANIPULATION
4546      ;WORK = BITS FROM SAVED CRC WORD (WCRC)
4547
4548 023514 012767 000022 155462 1$:  MOV      #18.,REPT1    ;GET 18 BITS PER WD
4549 023522 032767 000040 155414      BIT      #BITS,FLAG2    ;IN CRC TEST?
4550 023530 001403          BEQ      2$,          ;NO
4551 023532 012767 000020 155444      MOV      #16.,REPT1    ;YES
4552 023540 016767 155452 155432 2$:  MOV      WORK0,WCRC    ;SAVE CURRENT CRC WD
4553 023546 005067 155440          CLR      WK15         ;CLEAR BIT 15 FROM CRC AT T 1
4554 023552 000241          CLC                    ;CLEAR CARRY
4555 023554 006167 155436          ROL      WORK0       ;SHIFT CRC WD LEFT
4556 023560 006167 155426          ROL      WK15         ;CONTAINS BIT 15 OF CRC
4557 023564 032767 000040 155352      BIT      #BITS,FLAG2    ;IN CRC TEST?
4558 023572 001004          BNE      12$,        ;YES
4559 023574 022767 000021 155402      CMP      #17.,REPT1    ;DONE BITS 16 AND 17 YET?
4560 023602 101406          BLOS    3$,          ;NO
4561 023604 005067 155400          12$: CLR      INBIT       ;CLEAR WORK LOC
4562 023610 006104          ROL      R4           ;PUT DATA BIT FROM BUFFER
4563 023612 006167 155372          ROL      INBIT       ;IN WORK1 LOC
4564 023616 000402          BR      4$,          ;FOR BITS 16 AND 17
4565 023620 005067 155364          3$:  CLR      INBIT       ;GET BIT 15 OF CRC
4566 023624 016767 155362 155362 4$:  MOV      WK15,WORK    ;XOR BIT15 WITH INPUT BIT
4567 023632 004767 000220          5$:  JSR      PC,XXOR     ;TEST RESULT OF XOR
4568 023636 042767 000001 155352      BIC      #BIT0,WORK0
4569 023644 005767 155340          TST     INBIT
4570 023650 001403          BEQ      6$,          ;SAVE XOR RESULT OF BIT 0 AND INPUT
4571 023652 052767 000001 155336      BIS      #BIT0,WORK0
4572 023660 016767 155324 155270 6$:  MOV      INBIT,R50
  
```


N08

```

4573          ;FROM B0 IN WORK0 AND B1 IN SAVED CRC (WCRC) CALCULATE
4574          ;NEW B2 FOR WORK0
4575
4576 023666 005067 155322          CLR      WORK
4577 023672 032767 000002 155300  BIT      @BIT1,WCRC
4578 023700 001403          BEQ      7$
4579 023702 052767 000001 155304  BIS      @BIT0,WORK
4580 023710 016767 155242 155272 7$:  MOV      R50,INBIT
4581 023716 004767 000134          JSR      PC,XXOR
4582 023722 042767 000004 155266  BIC      @BIT2,WORK0
4583 023730 005767 155254          TST      INBIT          ;TEST RESULT OF XOR
4584 023734 001403          BEQ      8$
4585 023736 052767 000004 155252  BIS      @BIT2,WORK0
4586
4587          ;FROM B0 IN WORK0 AND B14 IN WCRC CLACULATE BIT15 IN WORK0
4588
4589 023744 005067 155244          8$:  CLR      WORK
4590 023750 032767 040000 155222  BIT      @BIT14,WCRC
4591 023756 001403          BEQ      9$
4592 023760 052767 000001 155226  BIS      @BIT0,WORK
4593 023766 016767 155164 155214 9$:  MOV      R50,INBIT
4594 023774 004767 000056          JSR      PC,XXOR
4595 024000 042767 100000 155210  BIC      @BIT15,WORK0
4596 024006 005767 155176          TST      INBIT          ;TEST RESULT OF XOR
4597 024012 001403          BEQ      10$
4598 024014 052767 100000 155174  BIS      @BIT15,WORK0
4599 024022 005367 155156          10$: DEC      REPT1          ;DONE WITH WD
4600 024026 001244          BNE      2$          ;NO
4601 024030 005367 155146          DEC      REPT          ;DONE WITH SECTOR?
4602 024034 001404          BEQ      11$          ;YES
4603 024036 062705 000002          ADD      @2,R5          ;GET NEXT WD
4604 024042 011504          MOV      (R5),R4          ;GET DATA WD
4605 024044 000623          BR       1$
4606 024046 016767 155144 155140 11$: MOV      WORK0,WORK          ;SAVE CRC WORD IN WORK
4607 024054 000207          RTS      PC          ;EXIT
4608
4609          ;XOR SUBROUTINE
4610
4611 024056 016703 155132          XXOR: MOV      WORK,R3
4612 024062 046703 155122          BIC      INBIT,R3
4613 024066 046767 155122 155114  BIC      WORK,INBIT
4614 024074 050367 155110          BIS      R3,INBIT
4615 024100 000207          RTS      PC
  
```

```

4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626 024102 010446
4627 024104 010546
4628 024106 017605 000004
4629 024112 032705 177400
4630 024116 001002
4631 024120 016605 000004
4632 024124 105715
4633 024126 001423
4634 024130 122715 000012
4635 024134 001012
4636 024136 116704 154653
4637 024142 116777 154646 154650
4638 024150 105777 154642
4639 024154 100375
4640 024156 005304
4641 024160 001370
4642 024162 112577 154632
4643 024166 105777 154624
4644 024172 100375
4645 024174 000753
4646 024176 017646 000004
4647 024202 062766 000002 000006
4648 024210 022666 000004
4649 024214 001006
4650 024216 062705 000002
4651 024222 042705 000001
4652 024226 010566 000004
4653 024232 012605
4654 024234 012604
4655 024236 000002
    
```

.SBTTL STYPE - TTY TYPEOUT ROUTINE

```

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

```

:TYPE: MOV R4,-(6) :SAVE R4
MOV RS,-(6) :SAVE RS
MOV @4(6),RS :GET ADDRESS TO BE TYPED
BIT @177400,RS :IS IT A TYPEN?
BNE 1$ :NO
MOV 4(6),RS :GET ADDRESS OF CHARACTER
1$: TSTB (RS) :TERMINATOR?
BEQ 2$ :GET OUT IF SO
CMPB @12,(RS) :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
MOVB (RS)+,@TPB :LOAD AND TYPE THE CHARACTER
4$: TSTB @TPS :IS THE PRINTER READY
BPL -4 :WAIT UNTIL IT IS
BR 1$ :GET THE NEXT CHARACTER
2$: MOV @4(6),-(6) :GET ADDRESS TO BE TYPED
ADD @2,6(6) :ADD 2 TO THE ADDRESS
CMP (6)+,4(6) :IS IT .+2?
BNE 3$ :NO
ADD @2,RS :ADD 2 TO THE ADDRESS
BIC @1,RS :BACK UP TO AN EVEN BYTE
MOV RS,4(6) :RESTORE ADDRESS
3$: MOV (6)+,RS :RESTORE RS
MOV (6)+,R4 :RESTORE R4
RTI :RETURN
    
```

```

4656          .SBTTL          SSCOPE - SCOPE LOOP HANDLER
4657
4658          ; THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
4659          ; LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
4660          ; "SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
4661          ; RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
4662
4663 024240 032737 000400 177570 .SCOPE: BIT      #SW8, @SWR      ; LOOP ON SPEC. TEST?
4664 024246 001404          BEQ      1S          ; NO LOOP ON SPEC. TEST
4665 024250 123767 177570 154522 CMPB   @SWR, ICNT   ; ON RIGHT TEST? #SW7-0#
4666 024256 001453          BEQ      .OVER      ; NOT RIGHT TEST
4667 024260 032737 040000 177570 1S:  BIT      #SW14, @SWR   ; LOOP ON TEST?
4668 024266 001045          BNE     .KIT        ; LOOP ON TEST IS SET
4669 024270 000416          BR       3S          ; SKIP - NOP FOR XOR TESTER
4670 024272 013746 000004          MOV     @R4, -(R6)  ; PUSH @R4 ON STACK
4671 024276 012737 024316 000004 MOV     @R5, @R4    ; SET FOR TIMEOUT
4672 024304 005737 177060          TST    @R17060    ; ERROR ON XOR?
4673 024310 012637 000004          MOV     (R6)+, @R4 ; POP STACK INTO @R4
4674 024314 000422          BR       .SVLAD    ; NO ERROR - GO TO NEXT TEST
4675 024316 022626          4S:  CMP     (R6)+, (R6)+ ; CLEAR STACK
4676 024320 012637 000004          MOV     (R6)+, @R4 ; POP STACK INTO @R4
4677 024324 000426          BR       .KIT        ; ERROR - LOOP ON TEST
4678 024326 032737 004000 177570 3S:  BIT      #SW11, @SWR   ; KILL ITERATIONS
4679 024334 001012          BNE     .SVLAD    ; YES - KILL ITERATIONS
4680 024336 105767 154437          TSTB   ICNT+1     ; FIRST ONE?
4681 024342 001404          BEQ     2S          ; BRANCH IF FIRST
4682 024344 126767 000060 154427 CMPB   TIMES, ICNT+1 ; DONE?
4683 024352 003013          BGT     .KIT        ; BRANCH IF NOT
4684 024354 112767 000001 154417 2S:  MOVB   @1, ICNT+1   ; FIRST ITERATION
4685 024362 105267 154412          .SVLAD: INCB   ICNT   ; COUNT TEST NUMBERS
4686 024366 011667 154416          MOV     (R6).LAD   ; SAVE LOOP ADDRESS
4687 024372 016737 154402 177570 MOV     ICNT, @DISPLAY ; DISPLAY TEST NO. AND ITERATION COUNT
4688 024400 000002          RTI                    ; RETURN
4689
4690 024402 105267 154373          .KIT:  INCB   ICNT+1   ; INC THE ITERATION COUNT
4691 024406 016737 154366 177570 .OVER: MOV     ICNT, @DISPLAY ; SET UP DISPLAY
4692 024414 005767 154370          TST    LAD         ; FIRST ONE?
4693 024420 001760          BEQ     .SVLAD    ; YES
4694 024422 016716 154362          MOV     LAD, (R6)  ; FUDGE RETURN ADDRESS
4695 024426 000002          RTI                    ; FIXES PS
4696
4697 024430 000001          TIMES: 1          ; RUN 1 TIMES
    
```

```

4698          .SBTTL          SHLT - HLT ROUTINE (ERROR TYPEOUT)
4699
4700          ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
4701          ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
4702          ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
4703          ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
4704          ;(HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADITIONAL TYPEOUTS.
4705
4706 024432 032737 002000 177570 .HLT: BIT      #SW10,2#SWR      ;BELL ON ERROR?
4707 024440 001402                BEQ      1$              ;NO - SKIP
4708 024442 104402                TYPE     BELL              ;RING BELL
4709 024446 005267 154330        1$: INC      ERRORS          ;COUNT THE NUMBER OF ERRORS
4710 024452 032737 020000 177570 BIT      #SW13,2#SWR      ;SKIP TYPEOUT IF SET
4711 024460 001025                BNE     2$              ;SKIP TYPEOUTS
4712 024462 104402 024466        TYPE     .+2              ;.ASCIZ <15><12>
4713 024472 011667 154314        MOV      (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
4714 024476 162767 000002 154306 SUB      #2,HLTADR       ;FUDGE ADDRESS
4715 024504 117767 154302 000066 MOVB    #HLTADR,.HLTCT  ;GET HLT ARGUMENT
4716 024512 016746 154274        MOV      HLTADR,-(6)    ;PUT HLTADR ON STACK
4717 024516 104404                TYPE0   TYPE           ;TYPE STACK IN OCTAL
4718 024520 104402 024524        TYPE     .+2              ;.ASCIZ " "
4719 024530 004767 001152        JSR     PC,RSREG        ;GO TO USER ERROR ROUTINE
4720 024534 005737 177570        2$: TST     2#SWR        ;HALT ON ERROR
4721 024540 100001                BPL     .+4              ;SKIP IF CONTINUE
4722 024542 000000                HALT    ;HALT ON ERROR!
4723 024544 032737 001000 177570 BIT      #SW9,2#SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
4724 024552 001010                BNE     4$              ;SKIP IF LOOP ON ERROR
4725 024554 105067 154221        CLRB   ICNT+1          ;CLEAR ITERATION COUNT
4726 024560 022737 021200 000042 CMP      #SENDAD,2#42   ;ACT11 AUTO ACCEPT?
4727 024566 001001                BNE     3$              ;NO BR
4728 024570 000000                HALT    ;
4729 024572 000002                3$: RTI     ;RETURN
4730 024574 000167 177602        4$: JMP     .KIT        ;LOOP ON TEST UNTIL NO ERRORS
4731
4732 024600 000000          .HLTCT: 0          ;HLT ARGUMENT
    
```

```

4733          .SBTTL          SOCTAL - OCTAL TYPEOUT ROUTINE
4734
4735          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
4736          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
4737          ;16 BITS. IT IS CALLED VIA THE TYOCT, TYPBIT, OR TYOCS MACRO'S.
4738
4739 024602 012767 170101 000160 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
4740 024610 000411              BR          .PTIT          ;NOW TYPE IT IN BIT FORM
4741 024612 112767 000001 000150 .TYPE0: MOVB   #1,.PR          ;SET ZERO FILL SWITCH
4742 024620 000402              BR          .+6          ;SKIP
4743 024622 005067 000142 .TYPES: CLR    .PR          ;SUPPRESS LEADING ZERO'S
4744 024626 112767 177772 000135 .MOVB   #8-6,.PR+1      ;SET COUNT
4745 024634
4746 024634 010446              MOV     R4,-(6)        ;PUSH R4 ON STACK
4747 024636 010546              MOV     R5,-(6)        ;PUSH R5 ON STACK
4748 024640 016605 000010      MOV     10(6),R5      ;GET THE DATA
4749 024644 012704 024772      MOV     #.PR+2,R4     ;SET POINTER TO FIRST ASCII CHAR.
4750 024650 105014              CLRB   (4)           ;CLEAR FIRST BYTE
4751 024652 000411              BR          .PRF       ;ROTATE FIRST BIT
4752 024654 105014 .PRL:  CLRB   (4)           ;CLEAR BYTE OF CHARACTER
4753 024656 032767 000100 000104 .BIT    #100,.PR      ;BIT TYPING MODE?
4754 024664 001004              BNE    .PRF          ;YES - SKIP 2 ROTATES
4755 024666 006105              ROL    R5            ;ROTATE BIT INTO C
4756 024670 106114              ROLB   (4)           ;PACK IT
4757 024672 006105              ROL    R5            ;ROTATE BIT INTO C
4758 024674 106114              ROLB   (4)           ;PACK IT
4759 024676 006105 .PRF:  ROL    R5            ;ROTATE BIT INTO C
4760 024700 106114              ROLB   (4)           ;PACK IT
4761 024702 105714              TSTB   (4)           ;IS IT ZERO?
4762 024704 001402              BEQ    .+6          ;SKIP INC
4763 024706 105267 000056      INCB   .PR          ;SET FILL SWITCH
4764 024712 105767 000052      TSTB   .PR          ;CHECK FILL SWITCH
4765 024716 001402              BEQ    .+6          ;SKIP BITSET
4766 024720 152724 000060      BISB   #'0,(4)+     ;MAKE INTO ASCII CHAR
4767 024724 105267 000041      INCB   .PR+1        ;INC COUNT
4768 024730 001351              BNE    .PRL          ;REPEAT
4769 024732 022704 024772      CMP    #.PR+2,R4     ;EMPTY BUFFER?
4770 024736 001002              BNE    .+6          ;SKIP IF NOT
4771 024740 112724 000060      MOVB   #'0,(4)+     ;LOAD 1 ZERO
4772 024744 105014              CLRB   (4)           ;NULL TERMINATOR
4773 024746 104402 024772      TYPE   .PR+2        ;TYPE IT
4774 024752 012605              MOV    (6)+,R5       ;POP STACK INTO R5
4775 024754 012604              MOV    (6)+,R4       ;POP STACK INTO R4
4776 024756 016666 000002 000004 .MOV    2(6),4(6)     ;GET RID OF
4777 024764 012616              MOV    (6)+,(6)     ;DATA WORD
4778 024766 000002              RTI                    ;RETURN
4779
4780 024770 000012 .PR:   .BLKW  12      ;COUNT, SWITCH, AND OUTPUT BUFFER

```

F09

M3INDEC-11-DERSD-C
DERSDC.SRC

17-MAY-77 14:16

RH70-RS04 MAINTENANCE MODE DIAGNOSTIC

MACY11 27(1006) 17-MAY-77 15:36 PAGE 110
SPOWER - POWER DOWN AND UP ROUTINES

```

4781          .SBTTL          SPOWER - POWER DOWN AND UP ROUTINES
4782
4783          ; THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
4784          ; THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
4785          ; WAIT FOR POWER TO GO DOWN AND BE RESTORED.
4786          ; IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
4787          ; THE PROGRAM WILL HALT AT '.ILLUP'.
4788
4789 025014 012777 025142 000126 .POWER: MOV      $.ILLUP, $.PUVEC ; SET FOR FAST UP
4790 025022 012777 000340 000122      MOV      $340, $.PUVECS+2 ; PRIO:7
4791 025030 010046          MOV      R0, -(6) ; PUSH R0 ON STACK
4792 025032 010146          MOV      R1, -(6) ; PUSH R1 ON STACK
4793 025034 010246          MOV      R2, -(6) ; PUSH R2 ON STACK
4794 025036 010346          MOV      R3, -(6) ; PUSH R3 ON STACK
4795 025040 010446          MOV      R4, -(6) ; PUSH R4 ON STACK
4796 025042 010546          MOV      R5, -(6) ; PUSH R5 ON STACK
4797 025044 010667 000076      MOV      SP, SAVR6 ; SAVE SP
4798 025050 012777 025060 000072      MOV      $.POWUP, $.PUVEC ; SET UP VECTOR
4799 025056 000000          HALT ; WAIT FOR PF
4800
4801 025060 016706 000062      .POWUP: MOV      .SAVR6, SP ; GET SP
4802 025064 005001          CLR      R1 ; WAIT LOOP FOR THE TTY
4803 025066 005201          IS: INC      R1 ; WAIT FOR THE INC
4804 025070 001376          BNE     IS ; OF WORD
4805 025072 012605          MOV      (6)+, R5 ; POP STACK INTO R5
4806 025074 012604          MOV      (6)+, R4 ; POP STACK INTO R4
4807 025076 012603          MOV      (6)+, R3 ; POP STACK INTO R3
4808 025100 012602          MOV      (6)+, R2 ; POP STACK INTO R2
4809 025102 012601          MOV      (6)+, R1 ; POP STACK INTO R1
4810 025104 012600          MOV      (6)+, R0 ; POP STACK INTO R0
4811 025106 012737 025014 000024      MOV      $.POWER, $.24 ; SET UP THE POWER DOWN VECTOR
4812 025114 012737 000340 000026      MOV      $340, $.26 ; PRIO:7
4813 025122 104402 025126          TYPE   .+2 ; .ASCIZ <15><12>"POWER"
4814 025136 000167 174054          JMP     MULSYS ; JMP TO USER ADDRESS
4815
4816 025142 000000          .ILLUP: HALT ; THE POWER UP SEQUENCE WAS STARTED
4817 025144 000776          BR      .-2 ; BEFORE THE POWER DOWN WAS COMPLETE
4818
4819 025146 000000          .SAVR6: 0 ; PUT THE SP HERE
4820 025150 000024 000026      .PUVEC: 24, 26 ; POWER UP VECTOR

```

```

4821          .SBTTL          SRDOCT - OCTAL INPUT ROUTINE
4822
4823          ; THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
4824          ; IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
4825
4826          025154 011646          .RDOCT: MOV      (6),-(6)          ; MOVE THE PC
4827          025156 016666 000004 000002 MOV      4(6),2(6)        ; MOVE THE PS
4828          025164 010146          MOV      R1,-(6)          ; PUSH R1 ON STACK
4829          025166 010246          MOV      R2,-(6)          ; PUSH R2 ON STACK
4830          025170 010346          MOV      R3,-(6)          ; PUSH R3 ON STACK
4831          025172 104412          4$:  RDLIN          ; READ A LINE INTO INPUT
4832          025174 005001          CLR      R1              ; INIT DATA WORD
4833          025176 012703 025376 MOV      8INPUT,R3       ; INIT POINTER
4834          025202 112302          1$:  MOVB      (3)+,R2     ; GET A BYTE
4835          025204 001417          BEQ      Z$              ; GET OUT IF ZERO
4836          025206 122702 000060 CMPB     8'0,R2          ; CHECK FOR 0 OR GREATER
4837          025212 003022          BGT      Z$              ; ERROR - LESS THAN 0
4838          025214 122702 000067 CMPB     8'7,R2          ; CHECK FOR 7 OR LESS
4839          025220 002417          BLT      Z$              ; ERROR - GREATER THAN 7
4840          025222 006002          ROR      R2              ; GET
4841          025224 006002          ROR      R2              ; INTO
4842          025226 006002          ROR      R2              ; POSITION
4843          025230 006101          ROL      R1              ; FIRST BIT
4844          025232 006102          ROL      R2              ; GET
4845          025234 006101          ROL      R1              ; SECOND BIT
4846          025236 006102          ROL      R2              ; GET
4847          025240 006101          ROL      R1              ; THIRD BIT
4848          025242 000757          BR       1$              ; LOOP
4849          025244 010166 000012 2$:  MOV      R1,12(6)        ; SAVE THE RESULT
4850          025250 012603          MOV      (6)+,R3        ; POP STACK INTO R3
4851          025252 012602          MOV      (6)+,R2        ; POP STACK INTO R2
4852          025254 012601          MOV      (6)+,R1        ; POP STACK INTO R1
4853          025256 000002          RTI                      ; RETURN
4854
4855          025260          3$:
4856          025260 104402 025264 TYPE     4.+2          ; .ASCIZ "'<15><12>
4857          025270 000740          BR       4$              ; TRY AGAIN
    
```

```

4858
4859
4860
4861
4862
4863
4864
4865 025272 010546
4866 025274 012705 025376
4867 025300 022705 025416
4868 025304 001412
4869 025306 105737 177560
4870 025312 100375
4871 025314 113715 177562
4872 025320 142715 000200
4873 025324 122715 000177
4874 025330 001005
4875 025332
4876 025332 104402 025336
4877 025342 000754
4878 025344 111527 000000
4879 025350 104402 025346
4880 025354 122725 000015
4881 025360 001347
4882 025362 105065 177777
4883 025366 104402 000012
4884 025372 012605
4885 025374 000002
4886
4887 025376 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      RS, -(6)          ;SAVE RS
1$:      MOV      @INPUT, RS      ;GET ADDRESS
2$:      CMP      @INPUT+16., RS  ;BUFFER FULL?
        BEQ      4$              ;YES - TYPE "?"
        TSTB     @177560         ;WAIT FOR
        BPL      -4              ;A CHARACTER
        MOVB     @177562, (5)    ;GET CHARACTER
        BICB     @200, (5)       ;GET RID OF JUNK
        CMPB     @177, (5)       ;IS IT A RUBOUT
        BNE      3$              ;SKIP IF NOT

4$:      TYPE     .+2             ;.ASCIZ "?"<15><12>
        BR       1$              ;ZAP THE BUFFER AND LOOP
3$:      MOVB     (5), #0        ;SET UP FOR TYPING
        TYPE     , 3$+2         ;ECHO IT
        CMPB     @15, (5)+      ;CHECK FOR RETURN
        BNE      2$              ;LOOP IF NOT RETURN
        CLRB     -1(5)          ;ZAP RETURN (THE 15)
        TYPE     , 12           ;TYPE A LINE FEED
        MOV      (6)+, RS       ;RESTORE RS
        RTI                    ;RETURN

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



```

4888
4889
4890
4891
4892
4893
4894
4895 025416 011646
4896 025420 162716 000002
4897 025424 017616 000000
4898 025430 062716 121036
4899 025434 013607
4900
4901 025436 024240
4902 025440 024102
4903 025442 024612
4904 025444 024622
4905 025446 025154
4906 025450 025272
4907 025452 025510
4908 025454 025536
4909 025456 021540
4910 025460 021510
4911 025462 021502
4912 025464 021604
4913 025466 021566
4914 025470 021632
4915 025472 022032
4916 025474 022126
4917 025476 023010
4918 025500 023240
4919 025502 022640
4920 025504 025630
4921 025506 025670
    
```

.SBTTL STRAP - TRAP HANDLER

```

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

```

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

```

4922                                     ;CLEAR ALL DISK REGISTERS
4923 025510 012777 000040 153364 .CLRDK: MOV #40,ARSCS2 ;CLEAR ALL DSK REG
4924 025516 016777 153436 153356 .MOV UNNUM,ARSCS2 ;GET UNIT NUMBER
4925 025524 005067 153444 .CLR MCCNT ;CLEAR MAINT CLOCK COUNT
4926 025530 005067 153442 .CLR MCCNT+2
4927 025534 000002 .RTI
4928
4929 025536 012777 000001 153360 .MRDMD: MOV #1,ARSMR ;PUT DRIVE INTO MAINT MODE
4930 025544 000002 .RTI
4931
4932 025546 005067 153442 WAITRY: CLR WORK ;CLEAR COUNTER
4933 025552 105777 153322 1S: TSTB ARSCS1 ;TEST READY
4934 025556 100406 .BMI 2S ;OK CONT
4935 025560 005267 153430 .INC WORK ;UPDATE COUNTER
4936 025564 005767 153424 .TST WORK ;DONE YET?
4937 025570 001403 .BEQ 3S ;READY DID NOT COME UP
4938 025572 000767 .BR 1S ;CONTINUE WAITING
4939 025574 062716 000002 2S: ADD #2,(SP) ;UPDATE RETURN PC
4940 025600 000207 3S: RTS PC ;RETURN
4941
4942                                     ;ROUTINE TO SHIFT COMPLETE DATA TABLE ONE BIT
4943                                     ;TO THE LEFT. CARRIES BIT 15 OF ONE WORD TO BIT 0 OF THE NEXT WORD
4944
4945 025602 012702 026610 MDATA: MOV #INBUF,R2 ;GET LEFT ADDRESS OF
4946 025606 062702 000442 .ADD #442,R2 ;DATA TABLE
4947 025612 012703 000220 .MOV #220,R3 ;SETUP COUNTER FOR 200 WORDS
4948 025616 000241 .CLC ;CLEAR CARRY
4949 025620 006142 1S: ROL -(R2) ;SHIFT DATA PATTERN
4950 025622 005303 .DEC R3 ;DO ALL
4951 025624 001375 .BNE 1S ;WORDS
4952 025626 000207 .RTS PC
4953
4954                                     ;THIS ROUTINE CLOCKS MR REG TO GET A SECTOR PULSE WHICH
4955                                     ;CLEARS OUT REGS. AND COUNTERS
4956
4957 025630 012767 002001 153344 .GETSP: MOV #1025.,REPT ;SETUP COUNTER
4958 025636 104430 .MRIND ;SEND INDEX PULSE TO MR REG
4959 025640 104422 1S: MRCLK ;CLOCK MR
4960 025642 005367 153334 .DEC REPT ;TO REACH
4961 025646 001374 .BNE 1S ;SECTOR PULSE
4962 025650 032777 000400 153246 .BIT #400,ARSMR ;DID SECTOR PULSE SET?????
4963 025656 001401 .BEQ 2S ;YES
4964 025660 000002 .RTI ;NO REPORT ERROR
4965 025662 062716 000002 2S: ADD #2,(SP) ;UPDATE RETURN ADDR
4966 025666 000002 .RTI
4967
4968 025670 104422 .SPASS: MRCLK ;CLOCK PAST SECTOR PULSE
4969 025672 104422 .MRCLK
4970 025674 005067 153274 .CLR MCCNT ;RESET MAINT CLOCK COUNTERS
4971 025700 005067 153272 .CLR MCCNT+2
4972 025704 000002 .RTI
    
```

```

;ERROR TYPTXTOUT ROUTINE
4973
4974
4975 025706 005767 176666 RSREG: TST .HLTCT ;SHOULD WE TYPTXT GOOD AND BAD
4976 025712 001022 BNE BS ;NO
4977 025714 104402 025720 TYPE .+2 ;ASCIZ "BAD="
4978 025726 010046 MOV BAD,-(6) ;PUT BAD ON STACK
4979 025730 104404 TYPE0 ;TYPE STACK IN OCTAL
4980 025732 104402 025736 TYPE .+2 ;ASCIZ "GOOD="
4981 025746 010146 MOV GOOD,-(6) ;PUT GOOD ON STACK
4982 025750 104404 TYPE0 ;TYPE STACK IN OCTAL
4983 025752 000402 BR BS ;TYPEOUT REGISTERS
4984 025754 000167 000432 JMP PTDONE ;GET OUT
4985 025760 BS:
4986 025760 104402 025764 TYPE .+2 ;ASCIZ "CS1="
4987 025772 017746 153102 MOV ARSCS1,-(6) ;PUT ARSCS1 ON STACK
4988 025776 104404 TYPE0 ;TYPE STACK IN OCTAL
4989 026000 15:
4990 026000 104402 026004 TYPE .+2 ;ASCIZ "ER="
4991 026012 017746 153076 MOV ARSER,-(6) ;PUT ARSER ON STACK
4992 026016 104404 TYPE0 ;TYPE STACK IN OCTAL
4993 026020 25:
4994 026020 104402 026024 TYPE .+2 ;ASCIZ "CS2="
4995 026032 017746 153044 MOV ARSCS2,-(6) ;PUT ARSCS2 ON STACK
4996 026036 104404 TYPE0 ;TYPE STACK IN OCTAL
4997 026040 032767 000200 176532 BIT #200,.HLTCT ;TYPTXT SECOND SET ?
4998 026046 001076 BNE SEEC ;YES
4999 026050 032767 000100 176522 BIT #AS,.HLTCT ;TYPTXT ER ?
5000 026056 001410 BEQ 3S ;NO
5001 026060 104402 026064 TYPE .+2 ;ASCIZ "AS="
5002 026072 017746 153020 MOV ARSAS,-(6) ;PUT ARSAS ON STACK
5003 026076 104404 TYPE0 ;TYPE STACK IN OCTAL
5004 026100 032767 000020 176472 3S: BIT #BA,.HLTCT ;TYPTXT BUS ADDRESS
5005 026106 001410 BEQ 4S ;NO
5006 026110 104402 026114 TYPE .+2 ;ASCIZ "BA="
5007 026122 017746 152760 MOV ARSBA,-(6) ;PUT ARSBA ON STACK
5008 026126 104404 TYPE0 ;TYPE STACK IN OCTAL
5009 026130 032767 000004 176442 4S: BIT #DA,.HLTCT ;TYPTXT DA ?
5010 026136 001410 BEQ 5S ;NO
5011 026140 104402 026144 TYPE .+2 ;ASCIZ "DA="
5012 026152 017746 152732 MOV ARSDA,-(6) ;PUT ARSDA ON STACK
5013 026156 104404 TYPE0 ;TYPE STACK IN OCTAL
5014 026160 032767 000010 176412 5S: BIT #WC,.HLTCT ;TYPTXT WC?
5015 026166 001410 BEQ 6S ;NO
5016 026170 104402 026174 TYPE .+2 ;ASCIZ "WC="
5017 026202 017746 152676 MOV ARSMC,-(6) ;PUT ARSMC ON STACK
5018 026206 104404 TYPE0 ;TYPE STACK IN OCTAL
5019 026210 032767 000040 176362 6S: BIT #DS,.HLTCT ;DRIVE STATUS
5020 026216 001475 BEQ PTDONE ;NO
5021 026220 104402 026224 TYPE .+2 ;ASCIZ "DS="
    
```

5022	026232	017746	152654			MOV	ARSDS,-(6)		;PUT ARSDS ON STACK
5023	026236	104404				TYPE0			;TYPE STACK IN OCTAL
5024	026240	000167	000146			JMP	PTDONE		;GET OUT
5025	026244	042767	000200	176326	SEEC:	BIC	#200,.HLTCT		;CLEAR COMMON BIT
5026	026252	032767	000240	176320		BIT	#DT,.HLTCT		;TYPTXT DRIVE TYPE?
5027	026260	001410				BEQ	9S		;NO
5028	026262	104402	026266			TYPE	.+2		;ASCIZ = DT="
5029	026274	017746	152626			MOV	ARSDT,-(6)		;PUT ARSDT ON STACK
5030	026300	104404				TYPE0			;TYPE STACK IN OCTAL
5031	026302	032767	000210	176270	9S:	BIT	#DB,.HLTCT		;TYPTXT DATA BUFFER
5032	026310	001410				BEQ	10S		;NO
5033	026312	104402	026316			TYPE	.+2		;ASCIZ = DB="
5034	026324	017746	152572			MOV	ARSD8,-(6)		;PUT ARSD8 ON STACK
5035	026330	104404				TYPE0			;TYPE STACK IN OCTAL
5036	026332	032767	000220	176240	10S:	BIT	#MR,.HLTCT		;TYPTXT MR?
5037	026340	001410				BEQ	11S		;NO
5038	026342	104402	026346			TYPE	.+2		;ASCIZ = MR="
5039	026354	017746	152544			MOV	ARSMR,-(6)		;PUT ARSMR ON STACK
5040	026360	104404				TYPE0			;TYPE STACK IN OCTAL
5041	026362	032767	000204	176210	11S:	BIT	#LA,.HLTCT		;TYPTXT LA?
5042	026370	001410				BEQ	PTDONE		;NO
5043	026372	104402	026376			TYPE	.+2		;ASCIZ = LA="
5044	026404	017746	152510			MOV	ARSLA,-(6)		;PUT ARSLA ON STACK
5045	026410	104404				TYPE0			;TYPE STACK IN OCTAL
5046	026412	052767	100000	152546	PTDONE:	BIS	#BIT15,ONCEE		;SET FORND ERROR FLAG
5047	026420	032767	000040	152540		BIT	#BITS,ONCEE		
5048	026426	001466				BEQ	1S		
5049	026430	104402	026434			TYPE	.+2		;ASCIZ <15><12>"MAINT CLOCK COUNT "
5050	026462	016767	152506	152536		MOV	MCCNT,WORK4		;GET MAINT CLOCK COUNT
5051	026470	016767	152502	152524		MOV	MCCNT+2,WORK2		;CAL NUMBERS FOR DOUBLE PRECISION
5052	026476	006167	152520			ROL	WORK2		
5053	026502	006167	152520			ROL	WORK4		
5054	026506	000241				CLC			
5055	026510	016746	152512			MOV	WORK4,-(6)		;PUT WORK4 ON STACK
5056	026514	104406				TYPES			;TYPE STACK IN OCTAL - SUPRESS
5057	026516	012767	000005	152504		MOV	#5,WORK5		
5058	026524	005067	152502		2S:	CLR	WORK6		
5059	026530	006167	152466			ROL	WORK2		
5060	026534	006167	152472			ROL	WORK6		
5061	026540	006167	152456			ROL	WORK2		
5062	026544	006167	152462			ROL	WORK6		
5063	026550	006167	152446			ROL	WORK2		
5064	026554	006167	152452			ROL	WORK6		
5065	026560	016746	152446			MOV	WORK6,-(6)		;PUT WORK6 ON STACK
5066	026564	104406				TYPES			;TYPE STACK IN OCTAL - SUPRESS
5067	026566	005367	152436			DEC	WORK5		
5068	026572	001354				BNE	2S		
5069	026574	104402	026600			TYPE	.+2		;ASCIZ <15><12>
5070	026604	000207			1S:	RTS	PC		
5071	026606	000000			FUDGE:	.WORD	0		;THIS WORD IS INSERTED TO INSURE INBUF ;STARTS ON BOUNDARY OF 4,10 SEE AIDS PR#2507.
5072									
5073									
5074	026610	000300			INBUF:	.BLKW	300		
5075	027410	000300			OUTBUF:	.BLKW	300		
5076		000001				.END			

B10

		2483	2487	2496	2500	2508	2512	2520	2524	2528	2534	2538	2547	2551
		2558	2572	2577	2608	2616	2620	2629	2633	2640	2646	2649	2652	2716
		2723	2732	2736	2745	2749	2756	2760	2766	2774	2779	2791	2793	2813
		2817	2821	2825	2835	2839	2849	2856	2859	2862	2881	2939	2945	2954
		2958	2967	2971	2979	2983	2989	2996	3001	3013	3015	3035	3039	3043
		3047	3057	3061	3071	3079	3082	3085	3151	3158	3167	3171	3180	3184
		3191	3195	3201	3209	3214	3226	3228	3244	3248	3252	3256	3266	3270
		3280	3288	3291	3349	3355	3364	3368	3377	3381	3388	3392	3398	3406
		3411	3423	3425	3441	3445	3449	3453	3463	3467	3477	3485	3488	3491
		3525	3537	3543	3547	3554	3566	3597	3604	3610	3616	3652	3656	3660
		3666	3670	3692	3697	3701	3707	3711	3765	3771	3780	3784	3793	3797
		3805	3809	3817	3821	3825	3831	3835	3844	3848	3855	3869	3874	3908
		3915	3919	3927	3931	3937	3939	3947	3950	3953	3956	4007	4016	4023
		4027	4039	4042	4045	4050	4053	4092	4095	4107	4110	4132	4136	
		765#	4713#	4714#	4715	4716	4733							
HLTADR	001012													
IADONE	016446	3591	3618#											
ICNT	001000	761#	900#	4163#	4665	4680	4682	4684#	4685#	4687	4690#	4691	4697	4725#
IE	= 000100	834#												
ILFDON	006734	1887	1889#											
INBIT	001210	855#	4561#	4563#	4565#	4569	4572	4580#	4583	4593#	4596	4612	4613#	4614#
INBUF	026610	2445	2462	2559	2594	2684	2783	2802	2868	2916	2933	3005	3024	3126
		3218	3234	3325	3415	3431	3741	3759	3856	3896	4083#	4085	4100	4124
		4125#	4539	4945	5074#									
INFTST	021130	4075	4140#											
INPUT	025376	910	4833	4866	4867	4887#								
INT	004604	1435#												
INTDON	004674	1444	1449#											
INTMR	020024	3757	3945#											
INTMR1	020074	3940	3957#											
IR	= 000100	820#												
LA	= 000204	811#	1481	1513	1581	1665	1672	5041						
LAD	001010	764#	901#	3583#	4162#	4686#	4692	4694	4697					
LBT	= 002000	828#												
LSTEV	001146	838#	4236#	4250#	4295									
LSTOD	001150	839#	4237#	4249#	4284									
LSTM0	023400	4498	4514#											
MCNT	001174	850#	1644#	4013#	4204#	4205#	4445#	4446#	4925#	4926#	4970#	4971#	5050	5051
MDATA	025602	3294	3497	4945#										
MPRO	= 172100	848#												
MR	= 000220	813#	1053	1110	1327	1334	1342	1350	1481	1513	1581	2468	2716	2939
		3151	3349	3765	5036									
MRAOE	020100	3968#												
MRCAL	023310	4300	4323	4496#										
MRCILF	006432	1854#												
MRCK	= 104420	1467	1473	1491	1495	1504	1508	1520	1524	1533	1537	1547	1551	1557
		1564	1568	1593	1597	1647	1652	1660	1667	1679	1683	1799	1825	1899
		1939	1999	2048	2053	2060	2064	2071	2075	2096	2105	2109	2118	2122
		2153	2204	2258	2307	2358	2434	2472	2481	2485	2494	2498	2506	2510
		2518	2522	2526	2532	2536	2545	2549	2556	2606	2614	2618	2627	2631
		2638	2671	2721	2730	2734	2743	2747	2754	2758	2764	2772	2777	2811
		2815	2819	2823	2833	2837	2847	2903	2943	2952	2956	2965	2969	2977
		2981	2987	2994	2999	3033	3037	3041	3045	3055	3059	3069	3105	3156
		3165	3169	3178	3182	3189	3193	3199	3207	3212	3242	3246	3250	3254
		3264	3268	3278	3312	3353	3362	3366	3375	3379	3386	3390	3396	3404
		3409	3439	3443	3447	3451	3461	3465	3475	3514	3585	3636	3684	3732
		3769	3778	3782	3791	3795	3803	3807	3815	3819	3823	3829	3833	3842

MRSRCH	007470	1996#																				
MRTIME	004676	1464#																				
MRTIM1	004744	1490#	1499																			
MRT2	005022	1512	1517#																			
MRT2A	005032	1519#	1528																			
MRT2B	005066	1532#	1541																			
MRT2C	005122	1546#	1555																			
MRT3	005240	1588#	1618																			
MRT4	005432	1641#																				
MRT4A	005504	1659#	1674																			
MRT4B	005562	1682#	1768																			
MRT4C	005614	1699#	1706																			
MRT4D	005666	1722#	1760																			
MRT4E	005674	1723#	1730	1772																		
MRT4F	006114	1735	1774#																			
MRT4G	006146	1786#																				
MRVR	020404	4073#																				
MRVRR	020714	4109	4115#																			
MRVR1	020544	4094	4097#																			
MRVR2	020720	2655	4118#																			
MRMCK	013364	2899#																				
MRMCK1	013540	2951#	2960																			
MRMCK2	013614	2976#	2985																			
MRMCK3	013716	3004#																				
MRMCK4	014116	3053#																				
MRMCK5	014162	3076#																				
MRMRT	011456	2430#	4115																			
MRMRT1	011624	2480#	2489																			
MRMRT2	011700	2505#	2514																			
MRMRT3	011764	2531#	2540																			
MRMR1	010066	2104#	2113																			
MULSYS	021216	4154	4160#	4814																		
MULTII	001672	906	911	943#	4166																	
N	= 000065	725#	1002	1006#	1066	1070#	1116	1120#	1137	1138#	1146	1150#	1175	1176#								
		1191	1192#	1203	1204#	1212	1213#	1228	1229#	1242	1243#	1249	1250#	1265								
		1266#	1279	1280#	1286	1287#	1299	1300#	1305	1306#	1311	1312#	1318	1319#								
		1328	1329#	1335	1336#	1343	1344#	1351	1355#	1367	1368#	1379	1380#	1391								
		1392#	1403	1407#	1431	1435#	1450	1454#	1628	1632#	1802	1806#	1890	1894#								
		1931	1935#	1984	1988#	2034	2038#	2141	2145#	2194	2198#	2246	2250#	2296								
		2300#	2343	2347#	2416	2420#	2656	2660#	2889	2893#	3086	3090#	3297	3301#								
		3500	3504#	3569	3573#	3619	3623#	3671	3675#	3713	3717#	3958	3962#	4066								
		4070#																				
NED	= 010000	823#	2396																			
NEDDON	011356	2372	2411#																			
NND0	011446	2404	2409	2412	2415#																	
NOPERR	021270	1909	1913	1924	1928	4171#																
NOMEV	001152	840#	4246#	4250	4259#	4261#	4269#	4271#	4292	4316	4420#	4426#	4428#	4435#								
		4437#	4488																			
NOMGO	002322	940	974	993	999#	4167																
NOMOD	001154	841#	4245#	4249	4256#	4258#	4266#	4268#	4281	4311	4419#	4423#	4425#	4432#								
		4434#	4485																			
ONCEE	001166	846#	903#	912#	956#	966	973#	975	981	999#	1465#	1602	1604#	1607#								
		1642#	1643#	1740	1745	1747#	1749#	1754#	1771#	1818#	1819	1841#	1845#	1850#								
		1852#	1860#	1862#	1867#	1870#	1897#	1903	1915#	1919#	1930#	1952	1979	1981#								
		1983#	1997#	2046#	2058#	2103#	2151#	2256#	2411	2414#	2431#	2432#	2479#	2566#								
		2567#	2578	2580	2585#	2590#	2591#	2668#	2669#	2728#	2796	2798#	2900#	2901#								

E10

		2950*	3018	3020*	3102*	3103*	3163*	3231	3233*	3309*	3310*	3360*	3428	3430*
		3581*	3588	3596*	3600*	3603*	3606*	3609*	3612*	3682*	3729*	3730*	3776*	3863*
		3864*	3875	3877	3882*	3887*	3888*	4081*	4164	4171	4173*	4234	4241*	4243
		4413	4453	4468	4497	4499	5046*	5047						
OR =	000200	821#												
OUTBUF =	027410	2705	2709	2869	3139	3143	3337	3341	3520	3550	3553	3632	3977	3980
		5075#												
PCNT =	001004	763#	4144*	4145*	4156									
PGE =	002000	822#												
PGTRAP =	004660	1436	1445#											
PIP =	020000	827#												
PS =	177776	727#	728	1006*	1071*	1438*	3727*							
PSM =	177776	728#												
PTDONE =	026412	4984	5020	5024	5042	5046#								
00 =	000001	1002#	1066#	1116#	1137#	1146#	1175#	1191#	1203#	1212#	1228#	1242#	1249#	1265#
		1279#	1286#	1299#	1305#	1311#	1318#	1328#	1335#	1343#	1351#	1367#	1379#	1391#
		1403#	1431#	1450#	1628#	1802#	1890#	1931#	1984#	2034#	2141#	2194#	2246#	2296#
		2343#	2416#	2656#	2889#	3086#	3297#	3500#	3569#	3619#	3671#	3713#	3958#	4066#
RBIT =	104444	2789	3011	3224	3421	4919#								
RDLIN =	104412	909	4831	4906#										
RDOCT =	104410	915	4905#											
REGCHG =	021400	4178	4183#											
REPT	001202	852#	1489#	1498#	1518#	1527#	1531#	1540#	1545#	1554#	1588#	1590#	1619#	1621#
		1658#	1673#	1698#	1705#	1721#	1759#	1770#	1774#	1781#	2012#	2017#	2057#	2067#
		2078#	2080#	2087#	2092#	2102#	2112#	2478#	2488#	2504#	2513#	2530#	2539#	2612#
		2621#	2727#	2737#	2752#	2761#	2786#	2794#	2808#	2831#	2840#	2870#	2878#	2949#
		2959#	2975#	2984#	3008#	3016#	3030#	3053#	3062#	3162#	3172#	3187#	3196#	3221#
		3229#	3239#	3261#	3271#	3359#	3369#	3384#	3393#	3418#	3426#	3436#	3458#	3468#
		3775#	3785#	3801#	3810#	3827#	3836#	3911#	3920#	3981#	3983#	3991#	3999#	4019#
		4028#	4086#	4090#	4101#	4105#	4455	4535#	4538#	4601#	4957#	4960#		
REPT1 =	001204	853#	1587#	1617#	1722#	1729#	1769#	2785#	2799	2803#	3007#	3021	3025#	3220#
		3417#	4464#	4466#	4470	4548#	4551#	4559	4599#					
RMRC1 =	010210	2150#												
RMRC2 =	010374	2202#												
RMRC3 =	010550	2255#												
RMRC4 =	010722	2305#												
RSAS =	001116	781#	933	935	950	1063	1946	1965	2027*	2028	2134*	2135	2318*	2392
		3518#	5002											
RSBA =	001106	777#	1009#	1033	1074*	1091	1178#	1179	1182*	1183	1186#	1187	1196#	1197
		1207#	1208#	1209	1393*	1395	1398#	1399	2462*	2705*	2933#	3139#	3337*	3520*
		3550	3552	3632*	3759#	3977*	4085#	4100*	4124*	5007				
RSBAB =	001142	791#	1394#											
RSCS1 =	001100	774#	1008#	1028	1073*	1086	1125*	1126	1129*	1130	1133*	1134	1140*	1141*
		1142	1357#	1359	1362*	1363	1439#	1446	1832*	1905#	1921#	1954#	1957#	2007#
		2024	2031	2082*	2131	2138	2157#	2164*	2178	2191	2208#	2229	2242	2262*
		2283	2293	2311#	2330	2340	2374#	2387	2464*	2644	2707*	2854	2935*	3077
		3141#	3286	3339#	3483	3522*	3523	3541	3593*	3607	3640*	3658	3695	3705
		3761#	3945	3986#	4087*	4088	4093	4102*	4103	4108	4126*	4127	4134	4933
		4987												
RSCS1B =	001134	788#	1358#	2395*										
RSCS2 =	001102	775#	929#	944#	948#	959#	1007*	1014*	1019	1022*	1072*	1079*	1081#	1082
		1085#	1152	1155#	1156	1158	1161#	1162	1165#	1166	1169#	1170#	1171	1174*
		1369#	1370#	1374	1413	2366*	2368	2376*	2380	2396	2400*	3532	3561	3687*
		4084#	4099#	4123#	4129	4923#	4924#	4995						
RSCS2B =	001136	789#	1371#											
RSDA =	001110	778#	1010#	1039	1075*	1097	1253*	1254	1257*	1258	1261*	1262	1271*	1272

G10

TST10	003376	1203#												
TST11	003422	1212#												
TST12	003504	1228#												
TST13	003536	1242#												
TST14	003562	1249#												
TST15	003644	1265#												
TST16	003676	1279#												
TST17	003722	1286#												
TST2	002566	1069#												
TST20	003764	1299#												
TST21	004006	1305#												
TST22	004030	1311#												
TST23	004054	1318#												
TST24	004112	1328#												
TST25	004140	1335#												
TST26	004176	1343#												
TST27	004234	1354#												
TST3	003002	1119#												
TST30	004306	1367#												
TST31	004352	1379#												
TST32	004422	1391#												
TST33	004474	1406#												
TST34	004602	1434#												
TST35	004674	1453#												
TST36	005430	1631#												
TST37	006212	1805#												
TST4	003074	1137#												
TST40	006734	1893#												
TST41	007124	1934#												
TST42	007466	1987#												
TST43	007650	2037#												
TST44	010206	2144#												
TST45	010372	2197#												
TST46	010546	2249#												
TST47	010720	2299#												
TST5	003122	1149#												
TST50	011134	2346#												
TST51	011446	2419#												
TST52	012374	2659#												
TST53	013362	2892#												
TST54	014220	3089#												
TST55	015100	3300#												
TST56	016030	3503#												
TST57	016250	3572#												
TST6	003262	1175#												
TST60	016446	3622#												
TST61	016762	3674#												
TST62	017102	3716#												
TST63	020076	3961#												
TST64	020402	4069#												
TST7	003344	1191#												
TTAGG	002570	1071#												
TYPE =	104402	908	914	931	955	968	972	989	990	1837	1859	1868	1949	1962
		1969	1976	2172	2173	2223	2224	2277	2278	2325	2413	2883	3492	3651
		3665	4114	4131	4148	4161	4174	4178	4341	4345	4347	4349	4351	4353
		4355	4357	4359	4361	4363	4365	4367	4369	4371	4373	4375	4708	4712

K10

MAINDEC-11-DERSO-C RH70-RSO4 MAINTENANCE MODE DIAGNOSTIC MACY11 27(1006) 17-MAY-77 15:36 PAGE 130
DERSDC.SRC 17-MAY-77 14:16 CROSS REFERENCE TABLE -- MACRO NAMES

. ABS. 030210 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DERSDC.SEQ/SOL/CRF/NL:TOC=DERSDC.SML,DERSDC.SRC/DS:ERFZ
RUN-TIME: 9 14 1 SECONDS
RUN-TIME RATIO: 540/25=21.4
CORE USED: 22K (43 PAGES)