

11/21+
RL01/02

RL01/02 DRIVE TEST 1
CNRLIAR

COPYRIGHT (c) 1977-83
AH-T748A-MC
FICHE 1 OF 1

APR 1984
digital
Made In USA

The image displays a grid of 150 small data tables arranged in 10 columns and 15 rows. Each table contains various data points, including numerical values, text labels, and possibly small diagrams or charts. The data is organized in a structured format, likely representing test results or system parameters. The text is small and difficult to read, but the overall layout is consistent across all tables.

11/21+
RL01/02

1
2
3 000000 000001
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
50
51
52

.TITLE CNRLIA RL01/2 DRIVE TEST 1
PART1==1
.ENABLE ABS
.LIST MC
.NLIST MD,ME,CND,TOC
.REM @

IDENTIFICATION

PRODUCT CODE: AC-T747A-MC
PRODUCT NAME: CNRLIA0 RL01/2 DRIVE TEST 1
PRODUCT DATE: DECEMBER 19, 1983
MAINTAINER: ISS DIAGNOSTIC SERVICES
AUTHOR: JAMES S. DOUCETTE

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

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

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

COPYRIGHT (C) 1983, DIGITAL EQUIPMENT CORPORATION

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

REVISION HISTORY

CHANGES MADE TO CZRLIDO IN PRODUCING CNRLIAO FOR THE SBC-11/21+ (FALCON-PLUS),
DEC. 19, 1983. CHANGES ARE IDENTIFIED BY ",JSD REV A".

1. CHANGED THE GENERAL OPERATING PRIORITY OF THE PROGRAM FROM LEVEL 7 TO LEVEL 6 TO ALLOW THE "BREAK" KEY TO INVOKE ODT. (THE TRAP HANDLER AND DEVICE INTERRUPT SERVICE ROUTINES STILL RUN BRIEFLY AT LEVEL 7). NOTE THAT HARD ERRORS MAY BE REPORTED BY THE DIAGNOSTIC IF THE USER EXECUTES A "BREAK-AND-PROCEED" SEQUENCE. HOWEVER, THE ERRORS WILL ONLY PERSIST FOR THE CURRENT PASS.
2. SET VECTOR 140 WITH THE ADDRESS OF ODT IN ROM (170000).
3. RE-DEFINED THE "WAITUS" AND "WAITMS" DELAY MACROS. ALSO CHANGED ALL "TIMDLY" MACRO CALLS TO "WAITUS", AND REMOVED ALL REFERENCES TO THE USE OF CLOCK INTERRUPTS FOR DELAY TIMING. UNDER FALCON-PLUS, CLOCK OPERATION IS NOT GUARANTEED. CLOCK INTERRUPTS MAY OR MAY NOT BE HARD-ENABLED, AND EVEN IF THEY WERE, THE INTERRUPT RATE COULD BE 50, 60, OR 800 HERTZ. FURTHERMORE, THE DRS "CLOCK" MACROS RETURN MISLEADING INFORMATION (UNDER FALCON-PLUS).
4. CHANGED THE WAIT TIME FOR COMPLETION OF A SEEK COMMAND (ROUTINE XSEEK) FROM THE AUTHOR'S ASSUMED 100 USEC (ACTUAL TIME WAS LONGER BECAUSE THE ORIGINAL VERSION OF THE "WAITUS" MACRO WAS GROSSLY INACCURATE) TO 800 USEC.

87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	CHAIN MODE OPERATION
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
3.1	ERROR REPORTING
3.1.1	SPECIFIC OPERATION MESSAGES
3.1.2	SPECIFIC RESULT MESSAGES
3.1.3	OTHER MESSAGES
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH CNDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER CNDP+, AND CAN BE CHAINED UNDER CNDP+, ACT AND APT IN ACT MODE (SEE 2.2 "CHAIN MODE OPERATION" FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC PROGRAM INTERFACES TO THE ENVIRONMENT AS IT EXECUTES. (IN THIS DOCUMENT, "CNDP+" REFERS TO THE FALCON-SPECIFIC XNDP+ SYSTEM).

WHEN THIS DIAGNOSTIC IS STARTED, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 6.25K OF THE UPPER PART OF MEMORY JUST BELOW THE CNDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

1.1.2 DIAGNOSTIC INFORMATION

THIS PROGRAM TESTS AND EXERCISES RL01/02 DISK DRIVES RL11/RLV11 CONTROLLERS (4 DRIVES PER CONTROLLER). THE ENTIRE PROGRAM IS RUN ON THE FIRST DRIVE BEFORE STARTING ON THE SECOND. THE PROGRAM STARTS BY TESTING THE SIMPLEST FUNCTIONS FIRST USING THE LOGIC TESTED IN EARLIER TESTS TO TEST MORE COMPLEX FUNCTIONS. THIS PROGRAM TESTS THE RL01/02 INTERFACE AND BASIC DRIVE LOGIC. GET STATUS WITH RESET, GET STATUS, SEEK, AND READ HEADER ARE THE ONLY COMMANDS EXECUTED IN THE PROGRAM. ONLY SEEKS WITH 0 DIFFERENCE ARE USED SO NO HEAD MOVEMENT IS REQUIRED. A SIGNIFICANT PORTION OF THE PROGRAM REQUIRES MANUAL INTERVENTION. THESE TESTS TEST THE COVER OPEN AND WRITE LOCK STATUS. THE DRIVE MUST BE LOADED AND UNLOADED TO TEST ALL THE CONDITIONS OF HEADS OUT, BRUSH HOME, AND DRIVE STATES. THE PROGRAM CAN BE RUN IN AUTOMATIC MODE IN WHICH CASE ALL TESTS REQUIRING MANUAL INTERVENTION ARE BYPASSED. WITHOUT MANUAL INTERVENTION, THE TEST REQUIRES APPROXIMATELY 80 SECONDS TO RUN.

187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

- * SBC-11/21+ PROCESSOR, 28KW MEMORY, JUMPERED FOR MEMORY MAP 0
- * CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- * 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
 - 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
 - 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- * CNDP+ (XXDP+) LOAD DEVICE (RL02, RX02, ETC.)
- * LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CNRLIA0 RL01/02 DRIVE TEST 1

1.3 RELATED DOCUMENTS AND STANDARDS

RL01/02 DISK SUBSYSTEM USER'S GUIDE (EK-RL012-UG-002)
XXDP+/USER'S MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLA	RLV11 RL01 DISKLESS TEST (RLV11 ONLY)
CNRLG	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CNRLH	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO

240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290

NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

THIS DIAGNOSTIC PROGRAM SHOULD BE LOADED AND STARTED USING NORMAL XXDP+ PROCEDURES. START THE EXECUTION OF THE CNDP+ MONITOR BY USING THE APPROPRIATE BOOTSTRAP PROGRAM. THE MONITOR WILL PRINT A MESSAGE IDENTIFYING ITSELF AND REQUESTING THAT THE CURRENT DATE BE ENTERED. AN EXAMPLE OF THIS MESSAGE IS GIVEN BELOW FOR THE CNDP+ MONITOR:

CNMDYAO CNDP+ DY MONITOR
BOOTED VIA UNIT 0
ENTER DATE (DD-MMM-YY):

AFTER THE DATE HAS BEEN ACCEPTED BY THE MONITOR THE RESTART ADDRESS OF THE MONITOR IS PRINTED. THEN THE FOLLOWING TWO QUESTIONS ARE ASKED:

50 HZ ? N
LSI ? N

THE DEFAULTS ARE BOTH "NO". TYPE "R" AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED THE FOLLOWING STEPS WILL OCCUR:

* STEP 1 *

THE DIAGNOSTIC WILL ISSUE THE PROMPT "DR>". FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART CNDP+, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT CNDP+. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO CNDP+ COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE CNDP+ "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE CNDP+ DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF

292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340

THESE ARE SET FORTH IN "2.3 DETAILS OF COMMANDS AND SYNTAX".
HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOME-
THING LIKE THIS:

STA/PASS:1/FLAGS:HOE

THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DR>" LEVEL NEED TO BE TYPED.
2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

PNT	PRINT NUMBER OF TEST BEING EXECUTED
LOE	LOOP ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

* STEP 2 *

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION "# UNITS?" TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE "HEADER" STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS "HEADER" STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392

* STEP 3 *

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE "HARDWARE QUESTIONS". THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED "HARDWARE P-TABLES". ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

* STEP 4 *

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR ALL THE UNITS, YOU WILL BE ASKED "CHANGE SW?" IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE "Y". IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE "N". IF YOU TYPE "Y" YOU WILL BE ASKED THE SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

* STEP 5 *

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DR>).
2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

LOE SET: THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.

NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURRED.

2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE RE-ISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN).
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED).
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURED. NO QUESTIONS ASKED).
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT

THE MOST TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY:

PRO/FLAGS:IER:LOE:HOE=0

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG

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

2. TURN ON THE LOE FLAG
3. TURN OFF THE HOE FLAG
4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER. WHEN YOU'VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE. YOU NOW HAVE THREE CHOICES:

1. START
2. RESTART
3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START
2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS (O-OPERATOR, D-DIAGNOSTIC):

491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546		BY WHOM ENTERED: -----
	.R NRLIAO	0
	DRS LOADED	D
	DIAG. RUN-TIME SERVICES REV. D APR-79	D
	CNRLI-A-0	D
	CNRLI TESTS THE RL01-02 INTERFACE	D
	AND BASIC DRIVE LOGIC	
	UNIT IS RL01, RL02	D
	DR>STA/PASS:1/FLAGS:HOE	D.0
	CHANGE HW (L) ? Y	D.0
	# UNITS (D) ? 2	D.0
	UNIT 0	D
	RL11 (L) Y ?	D.0
	BUS ADDRESS (0) 174400 ?	D.0
	VECTOR (0) 160 ?	D.0
	DRIVE (0) 0 ?	D.0
	DRIVE TYPE = RL01 (L) Y ?	D.0
	BR LEVEL (0) 5 ?	D.0
	UNIT 1	D
	RL11 (L) Y ?	D.0
	BUS ADDRESS (0) 174400 ?	D.0
	VECTOR (0) 160 ?	D.0
	DRIVE (0) 0 ? 1	D.0
	DRIVE TYPE = RL01 (L) ? N	D.0 (N=RL02)
	BR LEVEL (0) 5 ?	D.0
	CHANGE SW (L) ? N	D.0
	EXECUTE DRIVE SELECT TESTS (L) N ?	D.0
	EXECUTE HEAD ALIGNMENT SUPPORT (L) N ?	D.0
	DO MANUAL INTERVENTION TESTS (L) N ? Y	D.0
	INPUT ERROR LIMIT (D) 20 ?	D.0
	CNRLI HRD ERR 00004 TST 003 SUB 002 PC:004130	
	ERR HLT	
	DR>PRO/FLAGS:IER:LOE:HOE=0	D.0

	AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE	
	ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE	
	THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ↑C OUT	

	↑C	0

```

548
549 DR>CON/FLAGS:H0E:IER:LOE=0 D.0
550
551 CHANGE SW (L) ? N D.0
552
553 CNRLI EOP 1 D
554 ^C
555
556 DR>RESTART/PASS:1 D.0
557
558 CHANGE SW (L) ? N D.0
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

```

2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION. THE BIC FILES ARE CREATED BY USING THE SETUP UTILITY PROGRAM WHICH IS USED TO PARAMETERIZE THE DIAGNOSTIC PRIOR TO ITS EXECUTION. SETUP PROMPTS THE OPERATOR WITH THE HARDWARE AND SOFTWARE QUESTIONS. THE RESPONSE TO THESE QUESTIONS ARE USED TO BUILD P-TABLES. THE RESULT OF THE SETUP PROCESS IS A FILE WHICH INCLUDES THE DIAGNOSTIC WITH APPENDED P-TABLES. REFER TO THE XXDP+/SUPERVISOR USER'S MANUAL FOR A COMPLETE DESCRIPTION OF THE SETUP UTILITY.

TO RUN CHAIN MODE, THE CNDP+ MONITOR USES AN ASCII FILE (KNOWN AS A CHAIN FILE) LISTING THE PROGRAMS TO BE RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN. THIS FILE MUST BE ON THE SYSTEM DEVICE.

A CHAIN FILE MAY BE GENERATED BY USE OF THE XTECO TEXT EDITOR. THIS FILE MUST HAVE A CCC EXTENSION. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED BY THE CNDP+ MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENCOUNTERED.

TO EXECUTE A CHAIN FILE THE USER TYPES:

```

C FILNAM <CR> OR
C FILNAM/QV <CR>

```

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE CNDP+ MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PROGRAM COUNT IS NOT

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
643
644
645
646
647
648
649
650

USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OF QUICK VERIFY.

WHEN PROGRAMS ARE RUN IN CHAIN MODE, THE HARDWARE/SOFTWARE SWITCH REGISTERS SHOULD BE SET TO 000000. THE CNDP+ MONITOR PRINTS EACH COMMAND TAKEN FROM THE CHAIN FILE AND THEN EXECUTES THE COMMAND. WHEN THE LAST COMMAND OTHER THAN ANOTHER C COMMAND HAS BEEN EXECUTED THE CNDP+ MONITOR TERMINATES CHAIN MODE AND TYPES A PROMPT (.). IT IS READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE. IF THE LAST COMMAND IS ANOTHER C COMMAND, THE CHAIN MODE WILL CONTINUE AND THE CHAIN FILE SPECIFIED BY THIS NEW C COMMAND WILL BE USED.

IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERMINATION HE MAY DO SO BY TYPING A CONTROL/C. HOWEVER, THE MONITOR WILL NOT ABORT THE CHAIN MODE UNTIL IT RECEIVES PROGRAM CONTROL FROM THE PROGRAM CURRENTLY RUNNING.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

HOW ENTERED	LEGAL COMMANDS
-----	-----
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY

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
699
700
701
702

4. AN ERROR WAS ENCOUNTERED WITH THE MOE FLAG SET SET

FLAGS
ZFLAGS
EXIT

START
RESTART
CONTINUE
PROCEED
PRINT
DISPLAY
FLAGS
ZFLAGS
EXIT

2.3.2 COMMAND SYNTAX

STA(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED "RUN DIAGNOSTIC" B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH MOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

"TEST-LIST" IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5;8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

"PASS-CNT" IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING TEST EXECUTION. "FLAG-LIST" IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754

MOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUB-TEST, OR TEST) CONTAINING THE ERROR

IER INHIBIT ERROR REPORTING

IBE INHIBIT BASIC ERROR REPORTS

IXE INHIBIT EXTENDED ERROR REPORTS

PRI DIRECT ALL MESSAGES TO A LINE PRINTER

PNT PRINT NUMBER OF TEST BEING EXECUTED

BOE BELL ON ERROR

UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS

ISR INHIBIT STATISTICAL REPORTS

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC

ADR EXECUTE AUTODROP CODE

LOT LOOP ON TEST

EVL EVALUATE

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

"EOP-INCR" IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

RES(TART)/TEST;TEST-LIST/PASS;PASS-CNT/FLAGS;FLAG-LIST/EOP;EOP-INCR/
UNITS;UNIT-LIST

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW "P-TABLES" ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.

THE QUESTION "CHANGE SW?" IS ASKED AND THE ANSWERS GIVEN BECOME THE

756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806

NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. "UNIT-LIST" IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO "ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND". THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO "ALL") OR THE NEXT RESTART.
2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>

COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART
2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

PRO(CEED)/FLAGS:<FLAG-LIST>

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

EXIT

RETURN TO CNDP+ PROMPT MODE.

DRO(P)/UNITS:UNIT-LIST

THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED BACK OR UNTIL A START COMMAND IS GIVEN. A DROP CANNOT BE FOLLOWED BY A PROCEED.

THERE IS ALSO A "DROP" MACRO INTERNAL TO THE DIAGNOSTIC, WHICH GIVES THE FACILITY OF AUTO-DROPPING. THE DURATION OF A PROGRAM DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.

ADD/UNITS:UNIT-LIST

THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN PREVIOUSLY DROPPED BY THE DROP COMMAND) TO THE TEST SEQUENCE. AN ADD CANNOT BE FOLLOWED BY A PROCEED.

PRI(NT)

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

DIS(PLAY)/UNITS:<UNIT-LIST>

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

FLA(GS)

860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "N UNITS?" IS ANSWERED (WITH THE NUMBER N), SPACE IN CORE IS ALLOCATED FOR "N" P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT. IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT "BR LEVEL" 5. THE FIRST 4 DRIVES ARE RLO1'S AND THE LAST 4 DRIVES ARE RLO2'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 0
RL11 (L) Y ?
BUS ADDRESS (0) 174400 ?
VECTOR (0) 160 ?
DRIVE (0) 0 ? 0-3
DRIVE TYPE = RLO1 (L) Y ?
BR LEVEL (0) 5 ?

912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961

UNIT 4
RL11 (L) Y ?
BUS ADDRESS (0) 174400 ? 175400
VECTOR (0) 160 ? 164
DRIVE (0) 0 ? 0-3
DRIVE TYPE = RL01 (L) Y ? N
BR LEVEL (0) 5 ?

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE "BR LEVEL" (QUESTION #6). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #4 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11" TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #4 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. THE LAST QUESTION WAS DEFAULTED USING THE "BR LEVEL" FROM THE FIRST PASS.

2.5 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

RL11 (L) Y?

ANSWER YES(Y) IF YOU HAVE AN RL11 CONTROLLER, NO(N) IF YOU HAVE AN RLV11 CONTROLLER.

BUS ADDRESS (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (0) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

DRIVE (0) 0?

963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RLO1 (L) ?

ANSWER NO (N) IF DRIVE IS AN RLO2

BR LEVEL (0) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

CHANGE S.W. ?

A YES ANSWER WILL ASK THE FOLLOWING SOFTWARE PARAMETER QUESTIONS, WITH THE PRESENT DEFAULT VALUE PRINTED TO THE LEFT OF THE QUESTION MARK. (THE LAST ANSWER GIVEN IS THE DEFAULT) THE DEFAULT IS TAKEN ON A <CR>. CONTROL Z (↑Z) WILL DEFAULT ALL REMAINING QUESTIONS AND START THE TEST.

EXECUTE DRIVE SELECT TESTS (N)?

IF "YES" TESTS 5 AND 6 ARE EXECUTED IN THE FIRST PASS OF THE PROGRAM, THESE TESTS REQUIRE MANUAL INTERVENTION TO CHANGE ADDRESS PLUGS AND REQUIRE A FULL COMPLEMENT OF ADDRESS PLUGS (0 - 3).

EXECUTE HEAD ALIGNMENT SUPPORT (N)?

IF "YES", TEST 11 IS EXECUTED IN THE FIRST PASS.

EXECUTE MANUAL INTERVENTION TESTS (N)?

IF "YES", TESTS 1, 2, 3, AND 4 ARE EXECUTED TO TEST BASIC INTERFACE OPERATIONS, HEAD LOADING, HEAD UNLOADING, AND ALL STATE CHANGES.

SPECIFY ERROR LIMIT (DECIMAL) (20)?

THIS PARAMETER SPECIFIES THE MAXIMUM NUMBER OF ERRORS ALLOWED. THIS LIMIT IS ON A PER DRIVE BASIS IN A SINGLE PASS. IF THE ERROR LIMIT IS EXCEEDED, THE DRIVE IS DROPPED FROM FURTHER TESTING.

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

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVANT DATA.

3.1 ERROR REPORTING

MOST ERROR REPORTS HAVE THE FOLLOWING FORMAT.

- (1) PROG NAME ERR NUM TEST NUM SUBTEST NUM ERR PC
- (2) ROUTINE TRACE SEQ (IN SEQ CALLED)
(ADDRESS)
(ADDRESS)
.
(ADDRESS)
- (3) TEST DESCRIPTION
- (4) OPERATION:
- (5) RESULT:
- (6) ADDRESS OF UNIT UNDER TEST
- (7) RLCS RLDA RLBA RLMP CYL HD
- (8) OP INIT
- (9) OP DONE
- (10) DRIVE STATUS
- (11) WORD NUM IS (XXXXXX) SB (YYYYYY)
- (12) TOTAL COMPARE ERRS: (ZZZ) OF (128)

THE ONLY EXCEPTION TO THE ABOVE FORMAT IS PURE DATA COMPARE ERRORS (NOT DETECTED BY READ ERROR). THEN THE FORMAT DOES NOT INCLUDE LINES 5 THROUGH 10.

LINE 1 IS THE ERROR HEADER AND IS PROVIDED BY THE SUPERVISOR. THE PROGRAM IS IDENTIFIED BY NAME WITH THE NUMBER OF TEST AND SUB TEST PRESENTLY BEING EXECUTED.

THE SUBTEST NUMBER IS UNIQUE IN THIS PROGRAM IN THAT IT DOES NOT REFER TO A PHYSICAL SUBTEST WITHIN A GIVEN TEST. RATHER IT REFLECTS THE NUMBER OF TIMES A SUBTEST HAS BEEN EXECUTED WITHIN A TEST. CONSEQUENTLY, ON A TEST THAT TESTS AN INCREMENTAL TYPE OF OPERATION (SUCH AS INCREMENTAL SEEKS, READ ALL HEADERS FROM BOTH SURFACES, ETC.) THE SUBTEST WILL BE DESCRIPTIVE OF WHERE IN THE TEST THE ERROR OCCURRED.

THE ERROR P.C. IS THE PHYSICAL MEMORY LOCATION WHERE THE ERROR REPORT WAS INITIATED. SINCE MANY FUNCTIONS ARE SUBROUTINED, AND ERRORS ARE REPORTED FROM SUBROUTINES, THE ERROR P.C. IS NOT SUFFICIENT TO IDENTIFY THE LOCATION OF THE ERROR CALL AND THE ROUTINE TRACE SEQUENCE IS PROVIDED.

LINE 2 IS THE ROUTINE TRACE SEQUENCE. IF THE ERROR CALL IS INITIATED FROM

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

WITHIN THE TEST (AS OPPOSED TO WITHIN A ROUTINE), THIS PORTION OF THE REPORT IS OMITTED. IF THE CALL IS INITIATED FROM A ROUTINE (WHICH MAY BE CALLED BY ANOTHER ROUTINE, WHICH MAY BE CALLED BY ANOTHER ROUTINE, ETC. SEVERAL LEVELS DEEP) THE ROUTINE TRACE SEQUENCE PROVIDES A TRAIL TO THE ACTUAL LOCATION WITHIN THE TEST THAT CALLED THE FIRST ROUTINE. THE FIRST ENTRY LISTED IS THE LOCATION WHERE THE FIRST ROUTINE WAS CALLED.

LINE 3 IS THE TEST DESCRIPTION AND IS ROUGHLY IDENTICAL TO THE NAME OF THE TEST BEING PERFORMED.

LINE 4 IDENTIFIES THE ACTUAL HARDWARE FUNCTION THAT IS BEING PERFORMED. ADDITIONAL INFORMATION ON THIS LINE IS DESCRIPTIVE OF SPECIFIC USE OF THE FUNCTION. FOR EXAMPLE, THE OPERATION LINE WILL READ "READ HEADERS FOR 40 HEADERS" WHEN ALL HEADERS ARE BEING READ FROM A TRACK.

LINE 5 IDENTIFIES THE ERROR THAT HAS BEEN DETECTED. THE CONTENT OF LINE 5 IDENTIFIES WHAT WAS BEING TESTED (SUCH AS DRIVE READY, CONTROLLER ERROR, DRIVE STATE, ETC.), WHAT IT IS AND WHAT IT SHOULD BE. LINE 5 MAY BE REPEATED IF MORE THAN ONE TESTED ITEM IS FOUND IN ERROR.

IN ADDITION LINE 5 WILL REPORT ANY HARDWARE DETECTED ERRORS SUCH AS OPERATION INCOMPLETE, HEADER CRC, ETC. IN THIS CASE THE FIRST LINE PRINTED AS RESULT WILL BE DETERMINED BY THE THREE ERROR BITS OPI, HNF/DLT, AND HCRC/DCRC. THE LINE WILL BE DETERMINED AS IN THE FOLLOWING TRUTH TABLE:

HNF/DLT	DCRC/HCRC	OPI	MESSAGE
1	1	1	HDR NOT FND/HDR CRC/OPI ERROR
0	1	1	HDR CRC ERROR
1	0	1	HDR NOT FND ERROR
0	1	0	DATA CRC ERROR
1	0	0	DATA LATE ERROR

LINE 6 IDENTIFIES THE PHYSICAL ADDRESS OF THE UNIT UNDER TEST. THIS ADDRESS IS BY UNIBUS ADDRESS OF THE CONTROLLER AND DRIVE NUMBER.

LINE 7 NAMES THE CONTROLLER REGISTERS (AND CYLINDER AND HEAD WHERE THESE ARE APPLICABLE IN THE REPORT) TO BE REPORTED.

LINE 8 PROVIDES THE CONTENTS OF CONTROLLER REGISTERS WHEN THE OPERATION WAS INITIATED.

LINE 9 PROVIDES THE CONTENTS OF THE CONTROLLER REGISTERS WHEN THE ERROR BEING REPORTED WAS DETECTED. FREQUENTLY THE REGISTER CONTENTS OF OP INIT AND OP DONE WILL BE DIFFERENT. OP INIT MAY INDICATE A SEEK WAS BEING PERFORMED BUT OP DONE MAY INDICATE THE ERROR WAS DETECTED BY A READ HEADER. THE REASON IS THAT A SEEK WAS EXECUTED AND DID NOT PROPERLY POSITION HEADS AND WHEN THE READ HEADER WAS DONE THE HEADS WERE ON THE WRONG CYLINDER.

1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170

LINE 10 IS THE DRIVE STATUS. THIS LINE IS ONLY REPORTED IF THE RLMP REGISTER DOES NOT CONTAIN THE ACTUAL DRIVE STATUS.

LINE 11 AND LINE 12 ARE REPORTED IF THE ERROR WAS DETECTED AS A COMPARE OPERATION, EITHER DATA OR HEADERS. IN ADDITION, GOOD AND BAD DATA IS REPORTED FOR ALL READ ERRORS.

3.1.1 SPECIFIC OPERATION MESSAGES

THE OPERATION MESSAGE (LINE 4) IS GENERATED IN A DYNAMIC MANNER BASED ON THE SUBSYSTEM FUNCTION BEING EXECUTED AT THE TIME OF THE ERROR AND THE STATE OF THE FLAGS IN THE LOCATION TAGGED "OPFLAGS". THE POSSIBLE OPERATION MESSAGES ARE GIVEN BELOW.

SEEK -
FROM (CYL NUM) DIFF (CYL DIFF) SGN (0 OR 1) HD (0 OR 1) WHERE THE VALUES ARE GIVEN IN OCTAL. THIS MESSAGE IS THE RESULT OF A SEEK OPERATION THAT WAS VERIFIED BY A READ HEADER AND THE HEAD POSITION AFTER A SEEK IS IN ERROR. (THE ACTUAL HEAD POSITION IN THIS ERROR SITUATION IS GIVEN IN THE RESULT LINE, LINE 5.)

READ DATA -
IS A READ DATA OPERATION WHERE SOME FORM OF ERROR WAS DETECTED IN THE ACTUAL READ OPERATION. THIS ERROR COULD BE HARDWARE DETECTED SUCH AS DATA CRC, HEADER CRC, HEADER NOT FOUND, ETC., OR A SOFTWARE DETECTED ERROR SUCH AS DRIVE READY RESET AFTER A READ DATA COMPLETED.

READ DATA WITH DATA COMPARE -

IS AN ERROR THAT WAS DETECTED AS BAD DATA IN THE BUFFER AFTER A READ DATA OPERATION. WHEN THIS OPERATION IS REPORTED IT INDICATES THE ACTUAL READ DATA OPERATION COMPLETED WITH NO DETECTED ERRORS BUT THE DATA WAS WRONG.

READ HEADER -
READ HEADER FOR 40 HEADERS -
READ HEADER FOR 40 HEADERS WITH HEADER COMPARE -
HAVE THE SAME GENERAL MEANING AS THE READ DATA AND READ DATA WITH DATA COMPARE. MESSAGES HAVING THE OPERATION OF READ HEADER OR READ HEADER FOR 40 HEADERS ARE THE RESULT OF ERRORS DETECTED IN THE ACTUAL OPERATION WHILE THE READ HEADER FOR 40 HEADERS WITH HEADER COMPARE INDICATES NO ERROR IN THE ACTUAL OPERATION BUT THE HEADER DATA ITSELF WAS IN ERROR.

WRITE DATA -
RESET -
GET STATUS -
GET STATUS WITH RESET -
ARE ALL BASIC OPERATIONS. AS BEFORE, THE ERROR DETECTION CAN BE EITHER

1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223

HARDWARE OR SOFTWARE. THE RESULT LINE (LINE 5) WILL DEFINE THE REASON FOR THE REPORT.

LD DRV -
UNLD DRV -
ARE OPERATION MESSAGES THAT WILL APPEAR IN THE REPORT WHEN THE DRIVE LOAD AND UNLOAD SEQUENCE IS BEING TESTED.

ANOTHER GROUP OF OPERATION QUALIFIERS WILL BE REPORTED FOR OPERATIONS THAT FAIL IN SPECIFIC TESTS. THESE TESTS ARE THE WRITE/READ TEST PART 2, OVERWRITE TEST, AND THE ADJACENT CYLINDER INTERFERENCE TEST.

OPERATION -----	QUALIFIER -----
READ DATA WITH DATA COMPARE	FOL 0 TO CC SEEK
READ DATA	FOL 255 TO CC SEEK
WRITE DATA	FOL WRITE (NO SEEK)
READ HEADER	ADJ. CYL WRITTEN AFTER FWD SK
	ADJ. CYL WRITTEN AFTER REV SK
	SK FWD, WRT-SK REV, OVERWRT
	SK REV, WRT-SK FWD, OVERWRT

THE ABOVE OPERATIONS CAN BE REPORTED WITH ANY OF THE QUALIFIERS. THE QUALIFIERS IN THESE TESTS ARE AN ATTEMPT TO MAKE THE REPORT MORE MEANINGFUL BY PROVIDING INFORMATION ABOUT THE SEQUENCE OF OPERATIONS BEING DONE.

THE QUALIFIERS "FOL 0 TO CC SEEK" AND "FOL 255 TO CC SEEK" INDICATE THAT THE SEQUENCE OF OPERATIONS INCLUDED A SEEK OF A GIVEN DIRECTION TO THE CYLINDER WHERE THE TEST IS BEING PERFORMED.

THE "FOL WRITE (NO SEEK)" QUALIFIER MEANS THAT THE OPERATION WAS DONE AFTER A WRITE WITH NO HEAD MOVEMENT BETWEEN THE WRITE AND READ.

THE QUALIFIER "ADJ CYL WRITTEN AFTER FWD SK" AND "ADJ CYL WRITTEN AFTER REV SK" WILL BE REPORTED ONLY IN THE ADJACENT CYLINDER INTERFERENCE TEST. THESE QUALIFIERS ARE USED WHEN THE ERROR OCCURS ON THE CYLINDER UNDER TEST AND DEFINE THE DIRECTION THE HEADS WERE MOVED WHEN THE ADJACENT CYLINDER WAS WRITTEN.

THE QUALIFIERS "SK FWD, WRT-SK REV, OVERWRT" AND "SK REV, WRT-SK FWD, OVERWRT" WILL BE REPORTED ONLY IN THE OVERWRITE TEST. THESE QUALIFIERS DEFINE THE DIRECTION OF HEAD MOTION BEFORE THE INITIAL WRITE AND THE OVERWRITE.

THE QUALIFIER "ON BAD SEC FILES" WILL BE REPORTED WITH THE WRITE DATA COMMAND IF THE PROGRAM ABORTS THAT COMMAND BECAUSE THE WRITE WOULD BE ON THE BAD SECTOR FILES.

1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275

3.1.2 SPECIFIC RESULT MESSAGES

THE RESULT MESSAGE (LINE 5) IS GENERATED DYNAMICALLY BASED ON THE EXPECTED RESULT OF THE OPERATION BEING TESTED. SINCE OPERATIONS ARE MONITORED DURING EXECUTION THE RESULT MESSAGE MAY REPORT AN ERROR DETECTED DURING THE OPERATION AS WELL AS THE ERRORS SEEN AT THE END OF THE OPERATION. ONLY THE FIRST ERROR SEEN IS REPORTED IN ALL CASES.

THE GENERAL FORMAT FOR THE RESULT LINE IS:

RESULT:(VAR 1) IS (VAR 2) SB (VAR 3) (OPTIONAL QUALIFIER)
WHERE VARIABLE 1 CAN BE ONE OF THE FOLLOWING:

CONT ERR	(CONTROLLER ERROR)
DRV ERR	(DRIVE ERROR)
NON-EXSTNT MEM	(NON-EXISTENT MEMORY)
HDR CRC	(HEADER CRC ERROR)
DATA CRC	
HDR NOT FND	(HEADER NOT FOUND)
DATA LATE	
HDR NOT FND/HDR CRC/OPI	(ALL 3 BITS SET)
DRV RDY	(DRIVE READY)
SELECTED HEAD	
VOL CHK	(VOLUME CHECK)
COVER OPEN	
BRUSH HME	(BRUSH HOME)
WRT LCK	(WRITE LOCK)
HDS OUT	(HEADS OUT)
DRV SEL ERR	(DRIVE SELECT ERROR)
DRV STATE	(DRIVE STATE)
SPIN TIMEOUT	(SPINDLE TIMEOUT SPD ERROR)
WRT GAT ERR	(WRITE GATE ERROR)
SEEK TIMEOUT	(SKTO ERROR)
CUR HEAD ERR	(CURRENT IN HEAD ERROR)
WRT DAT ERR	(WRITE DATA ERROR)
OP INCOMPLETE	(OPI ERROR)
HDR/DAT ERR	(HDR CRC OR DATA CRC ERROR BIT 11 OF CS REGISTER)
	(HDR NOT FOUND OR DATA LATE ERROR BIT 12 OF CS REGISTER)
HDR NOT FND/DAT LATE	
CYL	(CYLINDER WHEN REPORTING A SEEK ERROR)

VARIABLE 2 WILL BE A VALUE THAT DEFINES WHAT THE RESULT ACTUALLY IS.

THIS CAN BE A 1 OR 0 TO INDICATE A SET OF RESULT CONDITIONS, A NUMBER 0 TO 7 TO INDICATE THE DRIVE STATE, OR A NUMBER 0 TO 377

1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327

(OCTAL) TO IDENTIFY A CYLINDER NUMBER.

VARIABLE 3 DEFINES THAT THE VALUE GIVEN IS VARIABLE 2 SHOULD BE. THE OPTIONAL QUALIFIER IS PROVIDED WHEN IT IS USEFUL TO KNOW WHEN THE ERROR WAS DETECTED IN THE OPERATION BEING PERFORMED. THIS QUALIFIER IS USED TO REPORT RESULTS SUCH AS:

BRUSH HME IS 1 SB 0 IN STATE 2
HEADS OUT IS 0 SB 1 IN STATE 3
DRV RDY IS 0 SB 1 IN DATA XFER
SELECTED HEAD IS 1 SB 0 IN CYCLE UP
DRV RDY IS 0 SB 1 IN STATE 5
DRV RDY IS 1 SB 0 IN SEEK W/O MOTION
DRV RDY IS 0 SB 1 IN 10MS
DRV RDY IS 0 SB 1 IN 500MS
DRV RDY IS 0 SB 1 IN 5SECONDS

THESE RESULTS, WHEN SEEN WITH THE OPERATION MESSAGE, WILL BE SELF EXPLANATORY.

OTHER RESULT MESSAGES THAT CAN BE PART OF AN ERROR REPORT ARE:

"INTERRUPT TOO LATE"

WHICH INDICATES THAT THE OPERATION BEING PERFORMED DID NOT COMPLETE IN THE EXPECTED AMOUNT OF TIME. THIS RESULT CAN BE CAUSED BY THE DRIVE LOSING READY BEFORE STARTING A READ HEADER AND THEREFORE NOT COMPLETING THE READ HEADER IN 1MS.

"FAIL TO RELOAD HEADS AFTER ERR CLEAR"

THIS IS REPORTED WHEN AN ERROR CAUSES HEADS TO UNLOAD AND AFTER THE ERROR IS CLEARED THE HEADS DO NOT RELOAD.

"UNKN DRV STATE-NO RDY, NO ERR, HDS OUT"

THIS IS REPORTED WHEN THE PROGRAM CANNOT DETERMINE THE DRIVE STATE OR STATUS.

"WRITE ABORTED"

THIS IS REPORTED WHEN THE PROGRAM ABORTS A WRITE TO PROTECT THE BAD SECTOR FILES.

"COULD NOT RETRIEVE DRIVE STATUS"

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

THIS IS REPORTED IF THE GET STATUS COMMAND DOES NOT COMPLETE SUCCESSFULLY WHEN THE STATUS IS REQUIRED TO REPORT AN ERROR.

"OPI SET-NO DRIVE RESPONSE"

THIS IS REPORTED AS THE RESULT WHEN THE GET STATUS COMMAND IS TIMED OUT (OPI SETS) WHEN THAT COMMAND IS BEING USED IN THE EARLY TESTS TO CHECK THE DRIVE INTERFACE.

"NO INTERRUPT ON CMND COMPLETE"

THIS IS REPORTED WHEN THE COMMAND SUCCESSFULLY COMPLETES BUT THE CONTROLLER HAS NOT GENERATED AN INTERRUPT.

"ERR DID NOT CLEAR"

THIS IS REPORTED WHEN THE RESET COMMAND DOES NOT CLEAR THE CONTROLLER ERRORS. THIS IS A CONTROLLER RELATED PROBLEM BUT IS REPORTED IF SEEN IN THE DRIVE TEST PROGRAMS.

"DRV ERR IS NOT CLEARED"

THIS IS REPORTED WHEN THE GET STATUS W/RESET COMMAND DOES NOT CLEAR ALL DRIVE ERRORS.

"UNEXPECTED ERR"

THIS IS REPORTED WHEN THE CONTROLLER SENSES AN ERROR BUT NO ERROR BITS ARE SET.

"BAD SEC FILE FMT ERR"

THIS IS REPORTED IF THE CONTENTS OF THE FILES DO NOT CORRESPOND TO THE EXPECTED FORMAT. (REFER TO DEC STANDARD 144 FOR FORMAT SPECIFICS.)

3.1.3 OTHER MESSAGES

OTHER INFORMATION IS REPORTED UNDER VARIOUS CIRCUMSTANCES. THESE ARE:

"BAD SEC FILES NOT STRD. ALL SEC ASSUMED GOOD."

THIS MESSAGE IS PRINTED WHEN A PARTICULAR TEST REQUIRES THE BAD

1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432

SECTOR FILES BUT THEY HAVE NOT BEEN STORED. THIS SITUATION WILL OCCUR IF THIS TEST IS STARTED OUT OF THE NORMAL PROGRAM SEQUENCE OR IF THE BAD SECTOR FILES COULD NOT BE READ.

"ERROR LIMIT EXCEEDED-UNIT DROPPED"

THIS IS REPORTED (WITH THE UNIT NUMBER) WHEN MORE THAN THE SPECIFIED NUMBER OF ERRORS (DEFAULT 20) HAVE OCCURED IN ANY SINGLE PASS.

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR CONTROL OF THE SUBSYSTEM.

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

- BIT 15 - COMPOSITE ERROR
- BIT 14 - DRIVE ERROR
- BIT 13 - NON EXISTANT MEMORY ERROR
- BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
- BIT 11 - HEADER CRC (WITH BIT 10 SET)
- DATA CRC (WITH BIT 10 CLEAR)
- BIT 10 - OPERATION INCOMPLETE
- BIT 9/8 - DRIVE SELECT (0-3)

1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484

BIT 7 - CONTROLLER READY
BIT 6 - INTERRUPT ENABLE
BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
BIT 4 - EXTENDED BUS ADDRESS (BIT 16)
BIT 3-1 - FUNCTION CODE
0 - NOP (PDP-11) MAINT (LSI-11)
1 - WRITE CHECK
2 - GET DRIVE STATUS
3 - SEEK
4 - READ HEADER
5 - WRITE DATA
6 - READ DATA
7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

BITS 15-1 BUS ADDRESS OF DATA TRANSFER
BIT 0 SHOULD BE 0

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (1-40.)

FOR SEEK FUNCTION

BIT 15-7 - DIFFERENCE TO NEW CYLINDER
BIT 6-5 - MUST BE ZERO (0)
BIT 4 - SURFACE (0=UPPER, 1=LOWER)
BIT 3 - MUST BE ZERO (0)
BIT 2 - SEEK DIRECTION(1=IN / 0=OUT)
BIT 1 - MUST BE ZERO (0)
BIT 0 - MUST BE ONE (1)

FOR GET STATUS FUNCTION

BIT 15-4 - IGNORED SHOULD BE ZERO (0)
BIT 3 - DRIVE RESET
BIT 2 - MUST BE ZERO (0)
BIT 1 - MUST BE ONE (1)

1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536

BIT 0 - MUST BE ONE (1)

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

BIT 15 - 0 - WORD COUNT (TWO'S COMPLEMENT)

FOR READ HEADER FUNCTION

BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
 - ZERO WORD (SECOND READ)
 - HEADER CRC (THIRD READ)

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (WL)
BIT 12 - SEEK TIME OUT (SKTO)
BIT 11 - SPIN ERROR (SPE)
BIT 10 - WRITE GATE ERROR (WGE)
BIT 9 - VOLUME CHECK (VC)
BIT 8 - DRIVE SELECT ERROR (DSE)
BIT 7 - DRIVE TYPE IS RLO2 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 BASIC INTERFACE TEST (PART 1)

1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587

LOAD IN DRIVE NUMBER. DO GET STATUS WITH RESET. IF OPI SETS:
DRIVE INTERFACE IS DEAD
DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
MARKER DETECTION FAILED
DRIVE IS NOT SELECTING OR AC LOW IS SET

SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT WITH NO OPI, CHECK STATUS RECEIVED. COVER OPEN
AND BRUSH HOME SHOULD BE SET. IF NOT:

BAD STATUS DATA LINE
BAD COVER SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER
BAD BRUSH HOME SWITCH OR LOGIC

CHECK WRITE LOCK STATUS BIT SET. IF NOT:
BAD SWITCH OR WRITE LOCK LOGIC
DRIVE COMMAND SHIFT REGISTER

CHECK STATE FOR 0. IF NOT:
BAD STATE ROM
DRIVE COMMAND SHIFT REGISTER

CHECK VOLUME CHECK RESET. IF NOT:
BAD RESET DETECTION
BAD VOLUME CHECK LOGIC
DRIVE COMMAND SHIFT REGISTER

CHECK DRIVE ERROR RESET. IF NOT:

BAD DRIVE ERROR INTERFACE
SOME OTHER ERROR STUCK ON. REPORT WHICH ERROR.

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2
IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED,
AND IS RUN IN FIRST PASS ONLY.

TEST 2 BASIC INTERFACE TEST (PART 2)

REQUEST OPERATOR TO CLOSE COVER AND RESET WRITE LOCK.

DO GET STATUS LOOP CHECKING IF COVER OPEN OR WRITE LOCK
RESETS. WAIT 15 SECONDS FOR BOTH TO CHANGE. IF NO CHANGE,
ASK OPERATOR TO TYPE CR IF PROCEDURE WAS FOLLOWED.

IF ONE CHANGED BUT NOT THE OTHER, REPORT WHICH FAILURE:

WRITE LOCK SWITCH OR LOGIC

1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639

(OR) COVER OPEN SWITCH OR LOGIC
DRIVE COMMAND SHIFT REGISTER

IF NEITHER CHANGED, REPORT BOTH FAILURES.

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2
IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED,
AND IS RUN IN FIRST PASS ONLY.

TEST 3 HEAD LOADING TEST

REQUEST OPERATOR TO PRESS LOAD SWITCH.

DO GET STATUS LOOP CHECKING FOR STATE TO GO TO 1. WAIT 30
SECONDS FOR CHANGE. IF NO CHANGE, ASK OPERATOR TO CONFIRM
ACTION BY TYPING CR.

IF LOAD WAS PRESSED:

BAD STATE ROM
BAD LOAD SWITCH OR LOGIC

CHECK THAT STATE 1 REMAINS FOR LESS THAN 30 SECONDS. IF NOT:

SPINDLE NOT TURNING OR TOO SLOW (AC SERVO)
SECTOR PULSE DETECTION OR LOGIC BAD
BAD CLOCK SHIFT REGISTER IN SPEED CONTROL
BAD DISK ON SPEED LOGIC
BAD STATE ROM

AND CHECK IF SPINUP TIMEOUT ERROR SET. IF NOT:

BAD STATE ROM
BAD TIMEOUT DETECTION LOGIC

CHECK THAT STATE GOES TO 2 OR 3 (WHICH STATE DEPENDS ON WHETHER
THE DRIVE HAS A BRUSH). IF NOT:

BAD STATE ROM

IF THE DRIVE HAS A BRUSH, CHECK THAT BRUSH HOME IS RESET 5
SECONDS OR LESS AFTER STATE IS 2. IF NOT:

BAD BRUSH HOME SWITCH OR LOGIC
BAD BRUSH MOTOR (AC SERVO)

WAIT 30 SECONDS FOR BRUSH HOME TO SET. IF NOT:

BAD AC SERVO

1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691

BAD SWITCH OR LATCH
CHECK THAT STATE HAS CHANGED TO 3. IF NOT:
BAD STATE ROM
AFTER STATE IS 3. CHECK HEADS OUT IS SET. IF NOT:
BAD SWITCH
BAD SEEK CONTROL ROM
BAD VELOCITY ROM
BAD DC SERVO

CHECK IF DRIVE ERROR IS SET. IF NOT:
BAD DRIVE ERROR LOGIC OR INTERFACE
WAIT 300 MS FOR STATE TO CHANGE TO 4. IF IT DOESN'T CHANGE:
STATE ROM BAD
SEEK ROM
VEL ROM
GUARD BAND DETECTION

WAIT 15 MS FOR STATE TO CHANGE TO 5.
CHECK VOLUME CHECK IS SET. IF NOT:
BAD VOLUME CHECK LOGIC
8 MS AFTER STATE GOES TO 5, DRIVE READY SHOULD SET. IF NOT:
INTEGRATOR OR NULL DETECTION FAILURE
READY ONE SHOT BAD
ENABLE TIMEOUT H NOT SETTING OR COUNT LOGIC BAD

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2 IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED, AND IS RUN IN FIRST PASS ONLY.

TEST 4 HEAD UNLOADING TEST

CHECK DRIVE IS READY. IF NOT REPORT AND ASK OPERATOR TO MAKE DRIVE READY.
REQUEST OPERATOR TO UNLOAD DRIVE.
LOOP ON GET STATUS WAITING FOR STATE TO CHANGE TO 6. IF NO CHANGE:

1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742

BAD STATE ROM
BAD SWITCH

WAIT 300 MS FOR STATE TO CHANGE TO 7. IF NO CHANGE:

BAD STATE ROM

AFTER STATE IS 7, WAIT 30 SEC FOR STATE TO CHANGE TO STATE 0.
IF NO CHANGE:

NO BRAKING
BAD AC SERVO

REQUEST OPERATOR TO LOAD DRIVE. WAIT UNTIL DRIVE BECOMES
READY.

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2
IS SELECTED, MANUAL INTERVENTION TESTING IS REQUESTED,
AND IS RUN IN FIRST PASS ONLY.

TEST 5 DRIVE SELECT TEST

INSTRUCT THE OPERATOR TO REMOVE DRIVE ADDRESS PLUGS FROM ALL
DRIVES EXCEPT THE DRIVE UNDER TEST. ASK THAT CARRIAGE RETURN
BE TYPED WHEN DONE.

DO GET STATUS TO ADDRESS OF DRIVE UNDER TEST. CHECK THAT NO
ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER ADDRESSES AND
CHECK THAT OPI SETS FOR ALL OTHER ADDRESSES.

DO GET STATUS TO ADDRESS OF NEXT SEQUENTIAL ADDRESS. CHECK
THAT NO ERRORS ARE REPORTED. DO GET STATUS TO ALL OTHER
ADDRESSES AND CHECK THAT OPI SETS.

REPEAT FOR ALL DRIVE ADDRESSES (0,1,2,3 - 0 IS SEQUENTIAL
AFTER 3).

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2
IS SELECTED, DRIVE SELECT TESTING IS REQUESTED, AND IS
RUN IN FIRST PASS ONLY.

TEST 6 DRIVE SELECT ERROR TEST

REQUEST OPERATOR INSERT IDENTICAL ADDRESS PLUGS IN TWO DRIVES
(MUST BE IDENTICAL TO NUMBER SPECIFIED EARLIER). REQUEST
OPERATOR TYPE CARRIAGE RETURN WHEN READY.

1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794

PROCEDURE WILL BE TO GET STATUS AND CHECK FOR DRIVE SELECT ERROR. THEN RESET THAT DRIVE AND VERIFY THAT DRIVE SELECT ERROR IS NOT REPORTED AGAIN. WAIT 1 SECOND, THEN CHANGE DRIVE SELECT TO A DIFFERENT NUMBER AND BACK AGAIN. DRIVE SELECT ERROR SHOULD SET AGAIN.

OPERATOR SHOULD SEE THE FAULT LIGHT ON ON BOTH DRIVES. IF INDICATOR IS NOT SEEN ON A DRIVE:

DRIVE SELECT ERROR DETECTION IS BAD IN THAT DRIVE.

NOTE: THIS TEST IS EXECUTED ONLY IF PROGRAM OPERATION MODE 2 IS SELECTED, DRIVE SELECT TESTING IS REQUESTED, AND IS RUN IN FIRST PASS ONLY.

TEST 7 INITIAL STATE TEST

INSTRUCT OPERATOR TO GO THROUGH A LOAD HEADS CYCLE TO INITIALIZE THE TEST.

DO GET STATUS, WAIT FOR INTERRUPT.

IF OPI OCCURS:

DRIVE INTERFACE IS DEAD
DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
DRIVE IS NOT SELECTING OR AC LOW IS SET
SYSTEM OR STATUS CLOCKS NOT OPERATIONAL
GET STATUS DETECTION FAILED.

IF INTERRUPT OCCURS WITHOUT OPI, CHECK DRIVE READY. READY SET INDICATES HEADS ARE LOADED AND ARE TRACKING (POSITION WORKING).

IF MANUAL INTERVENTION TESTS WERE RUN, CHECK THAT HEAD 0 IS SELECTED. IF NOT:

DRIVE CYCLE UP DID NOT SELECT HEAD 0

IF DRIVE READY IS SET, CHECK STATUS MESSAGE RECEIVED. HEADS OUT AND BRUSH HOME MUST BE SET. IF NOT:

DRIVE COMMAND SHIFT REGISTER NOT LOADING/SHIFTING
HEADS OUT OR BRUSH HOME SWITCH OR ASSOCIATED
CIRCUITRY BAD
STATUS DATA BAD

IF MANUAL INTERVENTION TESTS WERE RUN AND THIS IS THE FIRST

1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846

PASS CHECK THAT VOLUME CHECK AND DRIVE ERROR ARE SET.

CHECK ALL ERROR BITS ARE 0.

CHECK STATE IS 5. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD

NOTE: THIS TEST IS EXECUTED IF PROGRAM MODE 2 IS SELECTED,
MANUAL INTERVENTION TESTING IS REQUESTED, AND IS RUN
IN FIRST PASS ONLY.

TEST 8 INITIAL RESET STATE TEST

DO GET STATUS HEAD SELECT = 0, WAIT FOR INTERRUPT.

DO GET STATUS WITH RESET, WAIT FOR INTERRUPT. BOTH DRIVE
ERROR AND VOLUME CHECK SHOULD NOW BE RESET. IF NOT:

BAD RESET DETECTION, RESET ERROR, OR VOLUME CHECK FLOP

DRIVE COMMAND SHIFT REGISTER BAD

HEAD SELECTED BIT SHOULD STILL BE ZERO. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
HEAD SELECT SHIFT REGISTER NOT LOADING

NOTE: THIS TEST IS EXECUTED IF PROGRAM MODE 2 IS SELECTED,
MANUAL INTERVENTION TESTING IS REQUESTED, AND IS RUN
IN FIRST PASS ONLY.

TEST 9 DRIVE READY TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. WAIT FOR
INTERRUPT. GET STATUS. CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

CHECK DRIVE READY IS RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR
DOESN'T SET AT ALL:

HEADS MAY HAVE MOVED (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED

1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897

CHECK DRIVE ERROR DID NOT SET. IF IT SET, DO GET STATUS AND REPORT WHICH ERROR.

VERIFY HEAD SELECT IS ZERO.

TEST 10 SEEK SIGN SWITCH TEST

DO SEEK WITH DIFFERENCE 0, SIGN 1, HEAD 0. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

COUNT ROM
DIFFERENCE COUNTER PICKING UP BITS
COUNTER CIRCUITRY IS NOT INDICATING 0 DIFFERENCE

VERIFY DRIVE IS NOT READY

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
COUNT ROM

VERIFY DRIVE ERROR DID NOT SET

VERIFY HEAD SELECT IS ZERO.

DO SEEK WITH 0 DIFFERENCE, OPPOSITE SIGN, HEAD 0. REPEAT ABOVE TESTS.

TEST 11 HEAD ALIGNMENT SUPPORT ROUTINE

THIS TEST IS EXECUTED WHEN HEAD ALIGNMENT SUPPORT IS REQUESTED, AND IN THE FIRST PASS ONLY.

NOTE: THE NULL DETECTOR AND SEEK TIMEOUT SHOULD BE GROUND ON THOSE DRIVES WHICH LACK THE HEAD SELECT TEST POINTS. THE TEST WILL NOT SWITCH HEADS IF THERE IS A DRIVE FAULT.

THIS TEST SELECTS THE DRIVE UNDER TEST AND LOOPS ON A GET STATUS WITH RESET. THE WRITE LOCK BIT IS MONITORED AND WHEN WRITE LOCK IS RESET HEAD 0 IS SELECTED AND WHEN WRITE LOCK IS SET HEAD 1 IS SELECTED. THIS WILL PERMIT THE HEADS TO BE ALIGNED IN KEEPING WITH THE PRESENT HEAD ALIGNMENT PROCEDURE

1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948

WITHOUT RETURNING TO THE CONSOLE.

TYPING A CARRIAGE RETURN ON THE CONSOLE WILL TERMINATE THIS TEST ON THE DRIVE UNDER TEST. BEFORE TERMINATING, THE TEST WILL CHECK THAT WRITE LOCK IS RESET. IF NOT, THE OPERATOR WILL BE REQUESTED TO RESET WRITE LOCK.

TEST 12 HEAD SWITCHING TEST

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 1. WAIT FOR INTERRUPT. GET STATUS AND CHECK STATE IS 5. IF NOT:

DIFFERENCE COUNTER IS PICKING UP BITS
ASSOCIATED CIRCUITRY IS BAD

VERIFY DRIVE READY RESET. IF NOT:

ENABLE TIMEOUT OR READY LATCH/ONE SHOT BAD

WAIT APPROX 8 MS FOR READY TO SET. IF IT TAKES LONGER OR DOESN'T SET AT ALL:

HEADS ARE MOVING (INTEGRATOR OR NULL DETECTION)
READY ONE SHOT FAILED
DRIVE CANNOT TRACK WITH THIS HEAD

VERIFY DRIVE ERROR DID NOT SET.

DO GET STATUS, CHECK HEAD SELECT IS CORRECT. IF NOT:

HEAD SELECT REGISTER BAD
DRIVE COMMAND SHIFT REGISTER BAD

DO SEEK WITH 0 DIFFERENCE, SIGN 0, HEAD 0. REPEAT ABOVE TESTS.

TEST 13 READ HEADER TEST (PART 1)

DO SEEK WITH DIFFERENCE 0, HEAD 0, SIGN 0. WAIT FOR INTERRUPT AND WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT.

CHECK IF HEADER CRC ERROR SET. IF SET:

READ/WRITE BOARD BAD
READ DATA LINE BAD

1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

CHECK IF BIT 6 OF WORD 1 IS SAME AS HEAD SELECT BIT IN STATUS.
IF NOT:

HEADS ARE SWITCHED (CABLE)
HEAD SELECT LOGIC

IF MANUAL INTERVENTION TESTS WERE RUN AND HEAD ALIGNMENT TESTS
WERE NOT RUN, CHECK THAT HEADER WORD 0 INDICATES HEADS ARE
POSITIONED OVER CYLINDER 0. STORE HEADER WORD 1.

REPEAT TESTS USING HEAD 1.

CHECK THAT CYLINDER PORTION OF STORED HEADER WORD 1 IS THE
SAME AS HEADER WORD 1 OF THIS HEADER. IF NOT:

HEADS ARE MISALIGNED

TEST 14 READ HEADER TEST (PART 2)

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 0. WAIT FOR
INTERRUPT. WAIT FOR READY.

DO 40 CONSECUTIVE READ HEADER, STORE 3 HEADER WORDS AFTER EACH
READ.

CHECK ALL HEADERS FOR SEQUENCE AND CONTENT (WORD 2 ALL ZERO,
BIT 15 WORD 1 AND 3 IS 0, MS BIT WORD 1 IS 0). IF NOT:

BAD READ/WRITE BOARD
BAD PACK

DO SEEK WITH DIFFERENCE 0, SIGN 0, HEAD 1. REPEAT ABOVE TEST
FOR HEAD 1.

TEST 15 DIFFERENCE OF 1 SEEK TEST (PART 1)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.
DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED
HEADER WORD IS NOT 255 THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT
FOR INTERRUPT.
DO GET STATUS, WAIT FOR INTERRUPT. CHECK STATE IS 4. IF NOT:

DRIVE COMMAND SHIFT REGISTER BAD
DIFFERENCE REGISTER DROPPED BIT
STATE ROM FAILED

WAIT APPROX 5 MS. DO GET STATUS, WAIT FOR INTERRUPT. CHECK
STATE IS 5. IF NOT:

2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056

DIFFERENCE REGISTER NOT COUNTING
COUNT PULSE NOT GENERATED (COUNT LOGIC)
SEEK ROM FAILED
FAILURE IN DC SERVO
NO TACH FEEDBACK

WAIT APPROX 5 MS LONGER. TEST DRIVE READY. IF SET:
FAILURE IN READY LATCH OR INTEGRATOR

WAIT APPROX 5 MS LONGER. TEST READY. IF RESET:
FAILURE IN INTEGRATOR
UNEXPECTED GUARD BAND DETECTED

DO SEEK WITH DIFFERENCE 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

TEST 16 DIFFERENCE OF 1 SEEK TEST (PART 2)

DO READ HEADER, WAIT FOR INTERRUPT. STORE WORD 1 OF HEADER.

DO SEEK WITH DIFFERENCE OF 1, HEAD 0. IF CYLINDER OF STORED HEADER WORD IS NOT "MILIMIT" THEN SIGN BIT 1, ELSE SIGN BIT 0. WAIT FOR INTERRUPT, WAIT FOR DRIVE READY.

DO READ HEADER, WAIT FOR INTERRUPT. COMPARE CYLINDER OF THIS HEADER WITH CYLINDER OF STORED HEADER FOR DIFFERENCE OF ONE. IF NOT:

COUNT LOGIC BAD
INTEGRATOR FAILED

CHECK THAT HEADS MOVED FORWARD OR REVERSE AS EXPECTED. IF NOT:

SEEK ROM FAILED

DO SEEK WITH DIFFERENCE OF 1, OPPOSITE SIGN, HEAD 0. REPEAT ALL TESTS AS ABOVE.

REPEAT TEST USING HEAD 1.

NOTE: THIS TEST IS PERFORMED AT THE CYLINDER POSITION FOUND IN THE DRIVE WHEN THE TEST EXECUTES. CHOOSING A SINGLE SURFACE WILL LIMIT TESTING TO THAT SURFACE.

HEADER

2058		.SBTTL	HEADER
2059	002000		.=2000
2060			.MCALL SVC
2061			
2062	002000		SVC
2063	000001		SVCTST=1
2064	000001		SVCSUB=1
2065	000001		SVCBGL=1
2066	000000		SVCINS=0
2067	000000		SVCTAG=0
2068			
2069	002000	POINTER	BGNSW,BGNSFT,BGNDU
2070			
2071	002000	BGNMOD	MDHEDR
2072	002000	HEADER	CNRLI,A,0,1,0,PRI06
	002000		.ASCII /C/
	002001		.ASCII /N/
	002002		.ASCII /R/
	002003		.ASCII /L/
	002004		.ASCII /I/
	002005		.BYTE 0
	002006		.BYTE 0
	002007		.BYTE 0
	002010		.ASCII /A/
	002011		.ASCII /O/
	002012		.WORD 0
	002014		.WORD 1
	002016		.WORD L#HARD
	002020		.WORD L#SOFT
	002022		.WORD L#HW
	002024		.WORD L#SW
	002026		.WORD L#LAST
	002030		.WORD 0
	002032		.WORD 0
	002034		.WORD 0
	002036		.WORD 0
	002040		.WORD L#DISPATCH
	002042		.WORD PRI06
	002044		.WORD 0
	002046		.WORD 0
	002050		.BYTE C#REVISION
	002051		.BYTE C#EDIT
	002052		.WORD 0
	002054		.WORD 0
	002056		.WORD 0
	002060		.WORD L#DVTYP
	002062		.WORD 0
	002064		.WORD 0
	002066		.WORD 0
	002070		.WORD 0
	002072		.WORD L#DU
	002074		.WORD 0
	002076		.WORD L#DESC
	002100		EMT E#LOAD
	002102		.WORD 0
	002104		.WORD L#INIT
	002106		.WORD L#CLEAN

HEADER

```

002110 015654 .WORD L$AUTO
002112 014404 .WORD L$PROT
002114 000000 .WORD 0
002116 000000 .WORD 0
002120 000000 .WORD 0
2073 002122 ENDMOD
2074
2075 002122 DESCRIPT <CNRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC>
002122 103 116 122 .ASCIZ /CNRLI TESTS THE RL01-02 INTERFACE AND BASIC DRIVE LOGIC/
002125 114 111 040
002130 124 105 123
002133 124 123 040
002136 124 110 105
002141 040 122 114
002144 060 061 055
002147 060 062 040
002152 111 116 124
002155 105 122 106
002160 101 103 105
002163 040 101 116
002166 104 040 102
002171 101 123 111
002174 103 040 104
002177 122 111 126
002202 105 040 114
002205 117 107 111
002210 103 000

.EVEN
2076
2077 002212 DEVTYP <RL01,RL02>
002212 122 114 060 .ASCIZ #RL01,RL02#
002215 061 054 122
002220 114 060 062
002223 000

.EVEN
2078
2079 .SBTTL BIT AND OFFSET DEFINITIONS
2080
2081 002224 BGNMOD GLBEQAT
2082
2083 002224 EQUALS
;
; BIT DIFINITIONS
;
100000 BIT15== 100000
040000 BIT14== 40000
020000 BIT13== 20000
010000 BIT12== 10000
004000 BIT11== 4000
002000 BIT10== 2000
001000 BIT09== 1000
000400 BIT08== 400
000200 BIT07== 200
000100 BIT06== 100
000040 BIT05== 40
000020 BIT04== 20
000010 BIT03== 10

```

BIT AND OFFSET DEFINITIONS

```

000004 BIT02== 4
000002 BIT01== 2
000001 BIT00== 1
;
001000 BIT9== BIT09
000400 BIT8== BIT08
000200 BIT7== BIT07
000100 BIT6== BIT06
000040 BIT5== BIT05
000020 BIT4== BIT04
000010 BIT3== BIT03
000004 BIT2== BIT02
000002 BIT1== BIT01
000001 BIT0== BIT00
;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
; BIT POSITION IN SECOND STATUS WORD
000040 EF.START== 32. ; (100000) START COMMAND WAS ISSUED
000037 EF.RESTART== 31. ; (040000) RESTART COMMAND WAS ISSUED
000036 EF.CONTINUE== 30. ; (020000) CONTINUE COMMAND WAS ISSUED
000035 EF.NEW== 29. ; (010000) A NEW PASS HAS BEEN STARTED
000034 EF.PWR== 28. ; (004000) A POWER-FAIL/POWER-UP OCCURRED
;
; PRIORITY LEVEL DEFINITIONS
;
000340 PRI07== 340
000300 PRI06== 300
000240 PRI05== 240
000200 PRI04== 200
000140 PRI03== 140
000100 PRI02== 100
000040 PRI01== 40
000000 PRI00== 0
;
; OPERATOR FLAG BITS
;
000004 EVL== 4
000010 LOT== 10
000020 ADR== 20
000040 IDU== 40
000100 ISR== 100
000200 UAM== 200
000400 BOE== 400
001000 PNT== 1000
002000 PRI== 2000
004000 IXE== 4000
010000 IBE== 10000
020000 IER== 20000
040000 LOE== 40000
100000 HOE== 100000
;
; OFFSETS FOR HARDWARE P-TABLE
; CSR =0 ;BUS ADDRESS
; VECT =2 ;VECTOR ADDRESS

```

2084
2085
2086
2087

BIT AND OFFSET DEFINITIONS

2088	000004	PRIOR	=4	;PRIORITY
2089	000006	TYPDR	=6	;DRIVE TYPE
2090	000010	DRSB	=10	;DRIVE SELECT
2091	000012	CNT	=12	;CONTROLLER TYPE
2092				
2093				
2094	000000			
2095	000002			
2096	000004			
2097	000006			
2098	000010			
2099	000012			
2100				
2101				
2102	000001			
2103	000002			
2104	000004			
2105	000010			
2106	010000			
2107	020000			
2108	040000			
2109	100000			
2110				
2111				
2112	000102			
2113	000104			
2114	000106			
2115	000110			
2116	000112			
2117	000114			
2118	000116			
2119	000100			
2120				
2121				
2122	007777			
2123	000002			
2124	000001			
2125	000004			
2126	000010			
2127	000020			
2128	000040			
2129	000100			
2130	000200			
2131	000400			
2132	001000			
2133	002000			
2134	004000			
2135	010000			
2136	020000			
2137	040000			
2138	100000			
2139	003760			
2140				
2141				
2142				
2143	000001			
2144	000002			

```

PRIOR      =4
TYPDR     =6
DRSB      =10
CNT       =12

```

```

;PRIORITY
;DRIVE TYPE
;DRIVE SELECT
;CONTROLLER TYPE

```

; OFFSETS FOR SOFTWARE P-TABLE

```

MISWI     =0
LOLIM    =2
HILIM    =4
HEAD     =6
ERLIM    =10
DCLIM    =12

```

```

;SOFTWARE PARAMETERS SWITCHES
;CYLINDER LOWER LIMIT
;CYLINDER HIGH LIMIT
;SELECTED HEAD FOR RUNNING TESTS
;ERROR LIMIT
;DATA COMPARE ERROR LIMIT

```

; BIT ASSIGNMENTS FOR SOFTWARE P-TABLE SWITCHES

```

ALLCYL   =BIT00
ALLSEC   =BIT01
DRSELT   =BIT02
HDAALIGN =BIT03
HEADLM   =BIT12
MICYL    =BIT13
LOCYL    =BIT14
MITEST   =BIT15

```

```

;USE ALL CYLINDERS
;USE ALL SECTORS
;EXECUTE DRIVE SELECT TEST
;EXECUTE HEAD ALIGNMENT TEST
;HEAD LIMIT SPECIFIED FLAG
;HI LIMIT SPECIFIED FLAG
;LO LIMIT SPECIFIED
;EXECUTE MANUAL INTERVENTION TESTS

```

; SUBSYSTEM FUNCTIONS

```

CKDATA   =102
GTSTAT   =104
SEEK     =106
RDHEAD   =110
WTDATA   =112
RDDATA   =114
RDNOHR   =116
NOOP     =100

```

```

;WRITE CHECK
;GET STATUS
;SEEK
;READ HEADER
;WRITE DATA
;READ DATA
;READ DATA, IGNORE HEADERS
;NO OPERATION

```

; OPERATION FLAGS

```

COMPOP   =7777
HDRCMP   =BIT01
DATACMP  =BIT00
CYLUP    =BIT02
ULOAD    =BIT03
INOUTS   =BIT04
OUTINS   =BIT05
FOLWRT   =BIT06
REVSKS   =BIT07
FWDKSKS  =BIT08
REVSKO   =BIT09
FWDKSKO  =BIT10
BADADD   =BIT11
SEEKOP   =BIT12
RORWOP   =BIT13
RELDWT   =BIT14
HDR40    =BIT15
MQUALS   =OUTINS!INOUTS!FOLWRT!REVSKS!FWDKSKS!REVSKO!FWDKSKO

```

```

;COMPOSITE OPERATION FLAGS
;HEADER COMPARE OPERATION
;DATA COMPARE OPERATION
;CYCLE UP OPERATION
;UNLOAD OPERATION
;IN-OUT SEEK OPERATION
;OUT-IN SEEK OPERATION
;FOLLOWING WRITE OPERATION
;REV SEEK SEQ (ADJ INTERFERENCE)
;FWD SEEK SEQ (ADJ INTERFERENCE)
;REV SEEK SEQ (OVERWRITE)
;FWD SEEK SEQ (OVERWRITE)
;BAD DISK ADDRESS
;SEEK OPERATION
;READ OR WRITE OPERATION
;RELOAD WAIT
;40 HEADER OPERATION
;MESSAGE QUALIFIER BITS

```

; ERROR FLAGS FROM SUBROUTINES

```

TOSLOW   =BIT00
NOIRPT   =BIT01

```

```

;OPERATION TOOK TOO LONG
;NO INTERRUPT FROM OPERATION

```

BIT AND OFFSET DEFINITIONS

2145	000004	CONHNG	=BIT02	;CONTROLLER HUNG
2146	000010	NOCLR	=BIT03	;BAD CONTROLLER CLEAR
2147				
2148	000000	RLCS	=0	;CONTROL AND STATUS REGISTER
2149	000002	RLBA	=2	;BUS ADDRESS REGISTER
2150	000004	RLDA	=4	;DISK ADDRESS REGISTER
2151	000006	RLMP	=6	;MULTI-PURPOSE REGISTER
2152				
2153				
2154	000000	;	REGISTER BIT DEFINITIONS	- CONTROL STATUS REGISTER
2155	100000	RLCSR	=0	;CONTROL AND STATUS REGISTER
2156	040000	ANYERR	=100000	;ANY ERROR BIT
2157	020000	DRVERR	=40000	;DRIVE ERROR BIT
2158	010000	NXMERR	=20000	;NON-EXISTENT MEMORY ERROR
2159	010000	DLTERR	=10000	;DATA LATE ERROR
2160	004000	HNFERR	=10000	;HEADER NOT FOUND ERROR
2161	004000	DCKERR	=4000	;DATA CHECK ERROR
2162	004000	HRCERR	=4000	;HEADER CHECK ERROR
2163	002000	OPIERR	=2000	;OPERATION INCOMPLETE ERROR
2164	001400	DSMSK	=1400	;DRIVE SELECT MASK
2165	000200	CRDYMSK	=200	;CONTROLLER READY MASK
2166	000100	INTEBL	=100	;INTERRUPT ENABLE MASK
2167	000060	BAMSK	=60	;BUS ADDRESS UPPER MASK
2168	000001	DRDYMSK	=1	;DRIVE READY MASK
2169				
2170	000077	;	REGISTER BIT DEFINITIONS	- DISK ADDRESS FOR DATA XFER
2171	000100	SAMSK	=77	;SECTOR ADDRESS MASK
2172		HSMSK	=100	;HEAD SELECT MASK
2173				
2174	000001	;	REGISTER BIT DEFINITIONS	- DISK ADDRESS FOR SEEK
2175	000004	MBSET0	=1	;MUST BE SET, BIT 0
2176	000020	DIRBIT	=4	;DIRECTION BIT
2177		HDSEL	=20	;HEAD SELECT BIT
2178				
2179	000003	;	REGISTER BIT DEFINITIONS	- DISK ADDRESS FOR GET STATUS
2180	000010	GETSTAT	=3	;GET STATUS SETUP
2181		DRSET	=10	;DRIVE RESET MASK
2182				
2183	017777	;	REGISTER BIT DEFINITIONS	- MP FOR DATA XFER
2184	160000	WCMSK	=17777	;WORD COUNT MASK
2185		WCRNG	=160000	;WORD COUNT RANGE MASK
2186				
2187	000077	;	REGISTER BIT DEFINITIONS	- MP FOR READ HEADER
2188	000100	HDSEC	=77	;SECTOR MASK
2189		HDHSEL	=100	;HEAD SELECT MASK
2190				
2191	000007	;	REGISTER BIT DEFINITIONS	- MP FOR GET STATUS
2192	000010	STAMSK	=7	;STATE MASK
2193	000020	BHSTAT	=10	;BRUSH HOME STATUS
2194	000040	H0STAT	=20	;HEADS OUT STATUS
2195	000040	C0STAT	=40	;COVER OPEN STATUS
2196	000100	HSSTAT	=100	;HEAD SELECT STATUS
2197	000400	DSESTAT	=400	;DRIVE SELECT ERROR STATUS
2198	001000	VCSTAT	=1000	;VOLUME CHECK STATUS
2199	002000	WGESTAT	=2000	;WRITE GATE ERROR STATUS
2200	004000	SPDSTAT	=4000	;SPIN ERROR STATUS
2201	010000	ST0STAT	=10000	;SEEK TIMEOUT ERROR STATUS
2202	020000	WLSTAT	=20000	;WRITE LOCK STATUS

MACRO DEFINITIONS

```

2259 ;.MACRO TIMDLY ARG,?WAIT,?CLO,?CL1,?CL2,?CL3,?CL5
2260 ;
2261 ;     MOV     @ARG,DLYCNT     ;INITIALIZE DELAY COUNT
2262 ;
2263 ;     CMP     @1,CLKFLG      ;IF NO P-CLOCK, USE THE SOFTWARE LOOP, DELAY.
2264 ;     BLT     CL2
2265 ;
2266 ;     .NLIST
2267 ;     SETVEC  @104,@CLKINT,@340 ;SET P-CLOCK INTERRUPT VECTOR
2268 ;     .LIST
2269 ;     MOV     @1,@@172542     ;INITIALIZE CLOCK COUNT SET BUFFER REGISTER
2270 ;           ;/FOR 1 INTERRUPT PER 100 MICRO SECONDS
2271 ;     MOV     @113,@@172540   ;SET INTERRUPT ENABLE,REPEAT INTERRUPT MODE,
2272 ;           ;/10 KHZ RATE,START THE CLOCK
2273 ;WAIT: TST     DLYCNT         ;DELAY COUNT EXPIRED?
2274 ;     BNE     WAIT           ;BRANCH IF TIME NOT ELAPSED
2275 ;
2276 ;     CLR     @@172540        ;STOP THE P CLOCK
2277 ;     BR     CL5             ;SINCE P CLOCK EXISTS, AVOID SOFTWARE LOOP.
2278 ;
2279 ;CL2:  MOV     R2,-(SP)       ;SAVE R2 ON THE STACK.
2280 ;
2281 ;CLO:  MOV     LBASE,R2       ;PUT THE NUMBER OF SOFTWARE LOOPS INTO COUNTER
2282 ;
2283 ;CL1:  SUB     @1,R2          ;THIS IS THE SAME LOOP USED IN INITIALIZATION.
2284 ;     BLE     CL3            ;IT IS A SOFTWARE LOOP TO SIMULATE THE P CLOCK
2285 ;     BR     CL1            ;TIME IS APPROXIMATELY DLYCNT*100 US.
2286 ;
2287 ;CL3:  DEC     DLYCNT         ;
2288 ;     BGT     CLO           ;
2289 ;
2290 ;     MOV     (SP)+,R2        ;RESTORE R2 FROM THE STACK.
2291 ;CL5:  NOP
2292 ;.ENDM
2293
2294 ;.SBTTL GLOBAL DATA AND CONSTANTS
2295
2296 002224 BGNMOD GLBDAT
2297
2298 ;
2299 002224 000000 ; OPMSGs: TABLE OF OPERATION MESSAGES
2300 002226 005271 ;.WORD 0 ;FILLER
2301 002230 005315 ;.WORD MWRCHK ;MESSAGE FOR WRITE CHECK
2302 002232 005242 ;.WORD MGTSTA ; GET STATUS
2303 002234 005261 ;.WORD MSEEK ; SEEK
2304 002236 005303 ;.WORD MREADH ; READ HEADER
2305 002240 005250 ;.WORD MWRITE ; WRITE DATA
2306 002242 005400 ;.WORD MREAD ; READ DATA
2307 002244 005327 ;.WORD MWRSET ; WITH RESET
2308 002246 005346 ;.WORD MDATCP ; WITH DATA COMPARE
2309 002250 005445 ;.WORD MHDRCP ; WITH HEADER COMPARE
2310 002252 005434 ;.WORD MCYLUP ; LOAD HEADS
2311 002254 005476 ;.WORD MLOAD ; UNLOAD HEADS
2312 002256 005455 ;.WORD MINOUT ; IN-OUT SEQ
2313 002260 005521 ;.WORD MOUTIN ; OUT-IN SEQ
2314 002262 005543 ;.WORD MFOLWRT ; FOLLOWING WRITE
2315 002264 005576 ;.WORD MREVSK ; REV SEEK
; MFWSK ; FWD SEEK

```


GLOBAL DATA AND CONSTANTS

```

2316 002266 005665          .WORD  MRESKO          ;
2317 002270 005631          .WORD  MFWSKO          ;
2318 002272 005721          .WORD  MBADAD          ;
2319 002274 005364          .WORD  M40HDR          ;
2320 002276 000000          T.DRIVE: .WORD  0
2321 002300 000000          JJJ:      .WORD  0
2322 002302 000000          HLMTW:   .WORD  0
2323 002304 000000          CLRBYT: .WORD  0
2324 002306 000000          NXTML:  .WORD  0
2325 002310 000000          GBND:   .WORD  0
2326 002312 000000          CAMSK:  .WORD  0
2327 002314 000000          DIRMSK: .WORD  0
2328 002316 000000          HDCYL:  .WORD  0
2329
2330          ;
2331 002320 010415          ; RESTBL: TABLE OF RESULT NAME MESSAGE ADDRESSES
2332 002322 010526          .WORD  MCERR           ;CONTROLLER ERROR
2333 002324 011041          .WORD  MDRERR          ;DRIVE ERROR
2334 002326 011013          .WORD  MNEERR          ;NON-EXISTENT MEMORY ERROR
2335 002330 010776          .WORD  MFLERR          ;HEADER NOT FOUND-DATA LATE
2336 002332 010766          .WORD  MHDERR          ;HEADER OR DATA ERROR
2337 002334 011057          .WORD  MOPERR          ;OPERATION INCOMPLETE
2338 002336 000000          .WORD  MDRST          ;NO DRIVE STATUS AVAILABLE
2339 002340 010751          .WORD  0
2340 002342 010733          .WORD  MWDERR          ;WRITE DATA ERROR
2341 002344 000000          .WORD  MHCERR          ;HEAD CURRENT ERROR
2342 002346 010715          .WORD  0
2343 002350 010662          .WORD  MSTERR          ;SEEK TIMEOUT ERROR
2344 002352 010700          .WORD  MSPERR          ;SPINDLE ERROR
2345 002354 000000          .WORD  MMGERR          ;WRITE GATE ERROR
2346 002356 010632          .WORD  0
2347          .WORD  MDSERR          ;DRIVE SELECT ERROR
2348
2349          ;
2350 002360 004764          ; PATTBL: PATTERN TABLE
2351 002362 004766          .WORD  PAT1
2352 002364 005026          .WORD  PAT2
2353 002366 005066          .WORD  PAT3
2354 002370 005126          .WORD  PAT4
2355 002372 005134          .WORD  PAT5
2356 002374 005174          .WORD  PAT6
2357 002376 005176          .WORD  PAT7
2358 002400 005236          .WORD  PAT8
2359 002402 005240          .WORD  PAT9
2360          .WORD  PAT10
2361          ;
2362 002404 000000          ; SUBSTK: SUBROUTINE CALLING STACK
2363 002406 000000          .WORD  0                ;STACK IS 12 WORDS LONG
2364 002410 000000          .WORD  0
2365 002412 000000          .WORD  0
2366 002414 000000          .WORD  0
2367 002416 000000          .WORD  0
2368 002420 000000          .WORD  0
2369 002422 000000          .WORD  0
2370 002424 000000          .WORD  0
2371          .WORD  0
2372          ;RL01 TABLE OF CYLINDERS

```

GLOBAL DATA AND CONSTANTS

2373 002430 000002
 2374 002432 000006
 2375 002434 000011
 2376 002436 000014
 2377 002440 000021
 2378 002442 000026
 2379 002444 000033
 2380 002446 000042
 2381 002450 000051
 2382 002452 000200
 2383 002454 000377
 2384
 2385
 2386 002456 000004
 2387 002460 000014
 2388 002462 000022
 2389 002464 000030
 2390 002466 000042
 2391 002470 000054
 2392 002472 000066
 2393 002474 000104
 2394 002476 000122
 2395 002500 000400
 2396 002502 000777
 2397
 2398
 2399
 2400 002504
 2401 002544
 2402
 2403 002604 002
 2404 002605 007
 2405 002606 016
 2406 002607 024
 2407 002610 033
 2408 002611 041
 2409 002612 046
 2410 002613 055
 2411 002614 064
 2412 002615 072
 2413 002616 101
 2414 002617 110
 2415 002620 115
 2416 002621 124
 2417 002622 133
 2418 002623 141
 2419 002624 146
 2420 002625 154
 2421 002626 161
 2422 002627 170
 2423 002630 177
 2424 002631 206
 2425 002632 213
 2426 002633 222
 2427 002634 230
 2428 002635 235
 2429 002636 244

T25TBL: .WORD 2 ;TABLE OF DIFFERENCES
 .WORD 6
 .WORD 9.
 .WORD 12.
 .WORD 17.
 .WORD 22.
 .WORD 27.
 .WORD 34.
 .WORD 41.
 .WORD 128.
 .WORD 255.

;RL02 TABLE OF CYLINDERS

T25TB2: .WORD 4
 .WORD 12.
 .WORD 18.
 .WORD 24.
 .WORD 34.
 .WORD 44.
 .WORD 54.
 .WORD 68.
 .WORD 82.
 .WORD 256.
 .WORD 511.

; TABLE TO BE USED TO BUILD AND STORE THE CYLINDERS

T33TBL: .BLKW 16.
 TBT: .BLKW 16.

CYLTBL: .BYTE 2 ;TABLE OF DEFAULT CYLINDERS

.BYTE 7.
 .BYTE 14.
 .BYTE 20.
 .BYTE 27.
 .BYTE 33.
 .BYTE 38.
 .BYTE 45.
 .BYTE 52.
 .BYTE 58.
 .BYTE 65.
 .BYTE 72.
 .BYTE 77.
 .BYTE 84.
 .BYTE 91.
 .BYTE 97.
 .BYTE 102.
 .BYTE 108.
 .BYTE 113.
 .BYTE 120.
 .BYTE 127.
 .BYTE 134.
 .BYTE 139.
 .BYTE 146.
 .BYTE 152.
 .BYTE 157.
 .BYTE 164.

GLOBAL DATA AND CONSTANTS

2430	002637	252	.BYTE	170.
2431	002640	261	.BYTE	177.
2432	002641	270	.BYTE	184.
2433	002642	275	.BYTE	189.
2434	002643	303	.BYTE	195.
2435	002644	312	.BYTE	202.
2436	002645	317	.BYTE	207.
2437	002646	326	.BYTE	214.
2438	002647	334	.BYTE	220.
2439	002650	343	.BYTE	227.
2440	002651	352	.BYTE	234.
2441	002652	361	.BYTE	241.
2442	002653	367	.BYTE	247.
2443	002654	375	.BYTE	253.
2444	002655	000	.BYTE	0
2445	002656	000401	.WORD	257.
2446	002660	000406	.WORD	262.
2447	002662	000415	.WORD	269.
2448	002664	000423	.WORD	275.
2449	002666	000432	.WORD	282.
2450	002670	000445	.WORD	293.
2451	002672	000454	.WORD	300.
2452	002674	000463	.WORD	307.
2453	002676	000471	.WORD	313.
2454	002700	000500	.WORD	320.
2455	002702	000507	.WORD	327.
2456	002704	000514	.WORD	332.
2457	002706	000523	.WORD	339.
2458	002710	000532	.WORD	346.
2459	002712	000540	.WORD	352.
2460	002714	000545	.WORD	357.
2461	002716	000553	.WORD	363.
2462	002720	000560	.WORD	368.
2463	002722	000567	.WORD	375.
2464	002724	000576	.WORD	382.
2465	002726	000605	.WORD	389.
2466	002730	000612	.WORD	394.
2467	002732	000621	.WORD	401.
2468	002734	000627	.WORD	407.
2469	002736	000634	.WORD	412.
2470	002740	000643	.WORD	419.
2471	002742	000651	.WORD	425.
2472	002744	000660	.WORD	432.
2473	002746	000667	.WORD	439.
2474	002750	000674	.WORD	444.
2475	002752	000702	.WORD	450.
2476	002754	000711	.WORD	457.
2477	002756	000716	.WORD	462.
2478	002760	000725	.WORD	469.
2479	002762	000733	.WORD	475.
2480	002764	000742	.WORD	482.
2481	002766	000751	.WORD	489.
2482	002770	000760	.WORD	496.
2483	002772	000766	.WORD	502.
2484	002774	000774	.WORD	508.
2485	002776	000774	.WORD	508.
2486	003000	000000	.WORD	0

GLOBAL DATA AND CONSTANTS

2487	003002	000000	SSINDX: .WORD	0	;SUBROUTINE STACK INDEX POINTER
2488					
2489			; OPERATIONAL FLAGS		
2490	003004	000000	OPFLAG: .WORD	0	;OPERATION FLAGS
2491	003006	000000	DONE: .WORD	0	;OPERATION COMPLETE FLAG
2492	003010	000000	HADONE: .WORD	0	;HEAD ALIGNMENT DONE FLAG
2493	003012	000000	ERHEAD: .WORD	0	;ADDRESS OF ERROR HEADER
2494	003014	000000	MORECE: .WORD	0	;MORE THAN 1 COMPARE ERROR
2495	003016	000000	ERRSWI: .WORD	0	;ERROR RETURN SWITCH
2496	003020	000000	BSFLAG: .WORD	0	;BAD SECTOR FLAGS
2497	003022	000000	WRTSWI: .WORD	0	;WRITE SWITCH
2498	003024	000000	TBLSTR: .WORD	0	;TABLE STORAGE
2499					
2500	003026	000000	RLBAS: .WORD	0	;RL11 BASE ADDRESS
2501	003030	000000	RLVEC: .WORD	0	;RL11 VECTOR ADDRESS
2502	003032	000000	RLDRV: .WORD	0	;DRIVE NUMBER UNDER TEST
2503					
2504	003034	000000	L.CS: .WORD	0	;CONTROLLER REGISTER STORAGE
2505	003036	000000	L.BA: .WORD	0	;BEFORE OPERATION
2506	003040	000000	L.DA: .WORD	0	
2507	003042	000000	L.MP: .WORD	0	
2508	003044	000000	T.CS: .WORD	0	;CONTROLLER REGISTER STORAGE
2509	003046	000000	T.BA: .WORD	0	; AFTER OPERATION
2510	003050	000000	T.DA: .WORD	0	
2511	003052	000000	T.MP: .WORD	0	
2512	003052	000000	HDWRD1: .WORD	0	;HEADER WORD STORAGE
2513	003054	000000	HDWRD2: .WORD	0	
2514	003056	000000	HDWRD3: .WORD	0	
2515					
2516	003060	000000	T.STAT: .WORD	0	;DRIVE STATE STORAGE
2517					
2518	003062	000000	RESPARM: .WORD	0	;PARAM BLOCK FOR REASON REPORT
2519	003064	000000	.WORD	0	
2520	003066	000000	.WORD	0	
2521	003070	000000	.WORD	0	
2522	003072	000000	.WORD	0	
2523					
2524	003074	000000	DRVCNT: .WORD	0	;DRIVE COUNT FOR DRIVES UNDER TEST
2525	003076	000000	DIFAUG: .WORD	0	;DIFFERENCE ARGUMENT FOR SEEK
2526	003100	000000	OLDCYL: .WORD	0	;OLD CYLINDER
2527	003102	000000	NEWCYL: .WORD	0	;NEW CYLINDER
2528	003104	000000	CURCYL: .WORD	0	;CURRENT CYLINDER
2529	003106	000000	DESDIF: .WORD	0	;DESIRED DIFFERENCE
2530	003110	000000	DESSGN: .WORD	0	;DESIRED SIGN
2531	003112	000000	DESHD: .WORD	0	;DESIRED HEAD
2532	003114	000000	DESSEC: .WORD	0	;DESIRED SECTOR
2533	003116	000000	TEMP0: .WORD	0	;TEMPORARY STORAGE
2534	003120	000000	TEMP1: .WORD	0	;TEMPORARY STARGAGE
2535	003122	000000	TEMP2: .WORD	0	;TEMPORARY STORAGE
2536	003124	000000	TEMP3: .WORD	0	;TEMPORARY STORAGE
2537	003126	000000	TEMP4: .WORD	0	;TEMPORARY STORAGE
2538	003130	000000	TEMP5: .WORD	0	;TEMPORARY STORAGE
2539	003132	000000	TEMP6: .WORD	0	;TEMPORARY STORAGE
2540	003134	000000	TEMP7: .WORD	0	;TEMPORARY STORAGE
2541	003136	000000	TEMP8: .WORD	0	;TEMPORARY STORAGE
2542					
2577					

GLOBAL DATA AND CONSTANTS

2578	003140	000004	ERRVEC: .WORD	4	;ERROR VECTOR
2579	003142	000000	DLYCNT: .WORD	0	;DELAY COUNTER USED IN TIMING MACROS
2580	003144	000000	CLKFLG: .WORD	0	;FLAG INDICATING PRESENCE OF A L OR P CLOCK
2581	003146	000000	CLKADR: .WORD	0	;POINTER TO DIAGNOSTIC MONITOR CLOCK TABLE
2582	003150	000000	LBASE: .WORD	0	;L CLOCK ITERATION NUMBER TO FAKE P CLOCK
2583					
2584					
2585	003152	000000	; MISCELLANEOUS COUNTERS		
2586	003154	000000	PASCNT: .WORD	0	;PASS COUNTER (LOCAL TO A TEST)
2587	003156	000000	COUNT: .WORD	0	;A COUNTER (LOCAL TO A TEST)
2588	003160	000000	ERRPOINT: .WORD	0	;ERROR POINTER
2589	003360	000000	ERRCNT: .BLKW	64.	;ERROR COUNTER FOR PROGRAM
2590	003362	000000	PASNUM: .WORD	0	;PASS NUMBER FOR PROGRAM
2591	003364	000	PSETNM: .WORD	0	;COUNTER FOR PARAMETER SET NUMBER IN USE
2592	003365	000	LOCERR: .BYTE	0	;LOCAL ERROR COUNTER
2593	003366	000000	NOERCT: .BYTE	0	;INHIBIT ERROR COUNTING FLAG
2594	003370	000000	TRPFLG: .WORD	0	;HARDWARE TRAP FLAG
2595			PWRFLG: .WORD	0	;POWER FAILURE FLAG
2596					
2597	003372	000000	; BAD SECTOR TABLES AND POINTERS		
2598			BSFVAL: .WORD	0	;BAD SECTORS FILES VALID FLAG
2599	003374		SBSFIL: .BLKW	76	;SOFTWARE BAD SECTOR FILE
2600	003570		FBSFIL: .BLKW	76	;FACTORY BAD SECTOR FILE
2601					
2602	003764		IBUFF: .BLKW	200	;INPUT BUFFER
2603	004364		OBUFF: .BLKW	200	;OUTPUT BUFFER
2604					
2605	004764	000000	PAT1: .WORD	0	;PATTERN 1 (ALL ZEROS)
2606	004766	177772	PAT2: .WORD	177772	
2607	004770	177777	.WORD	177777	
2608	004772	177777	.WORD	177777	
2609	004774	052525	.WORD	052525	
2610	004776	052525	.WORD	052525	
2611	005000	052525	.WORD	052525	
2612	005002	177777	.WORD	177777	
2613	005004	177777	.WORD	177777	
2614	005006	052525	.WORD	052525	
2615	005010	052525	.WORD	052525	
2616	005012	177777	.WORD	177777	
2617	005014	052525	.WORD	052525	
2618	005016	177252	.WORD	177252	
2619	005020	177252	.WORD	177252	
2620	005022	172765	.WORD	172765	
2621	005024	172765	.WORD	172765	
2622					
2623	005026	000003	PAT3: .WORD	000003	
2624	005030	000000	.WORD	000000	
2625	005032	000000	.WORD	000000	
2626	005034	177777	.WORD	177777	
2627	005036	177777	.WORD	177777	
2628	005040	177777	.WORD	177777	
2629	005042	000000	.WORD	000000	
2630	005044	000000	.WORD	000000	
2631	005046	177777	.WORD	177777	
2632	005050	177777	.WORD	177777	
2633	005052	000000	.WORD	000000	
2634	005054	177777	.WORD	177777	

GLOBAL DATA AND CONSTANTS

2635	005056	000000		.WORD	000000
2636	005060	177777		.WORD	177777
2637	005062	000000		.WORD	000000
2638	005064	177777		.WORD	177777
2639					
2640	005066	025252	PAT4:	.WORD	025252
2641	005070	052525		.WORD	052525
2642	005072	052525		.WORD	052525
2643	005074	125252		.WORD	125252
2644	005076	125252		.WORD	125252
2645	005100	125252		.WORD	125252
2646	005102	052525		.WORD	052525
2647	005104	052525		.WORD	052525
2648	005106	125252		.WORD	125252
2649	005110	125252		.WORD	125252
2650	005112	052525		.WORD	052525
2651	005114	125252		.WORD	125252
2652	005116	052525		.WORD	052525
2653	005120	125252		.WORD	125252
2654	005122	052525		.WORD	052525
2655	005124	125252		.WORD	125252
2656					
2657	005126	155555	PAT5:	.WORD	155555
2658	005130	133333		.WORD	133333
2659	005132	066666		.WORD	066666
2660					
2661	005134	121105	PAT6:	.WORD	121105
2662	005136	150442		.WORD	150442
2663	005140	064221		.WORD	064221
2664	005142	132110		.WORD	132110
2665	005144	055044		.WORD	055044
2666	005146	026442		.WORD	026442
2667	005150	013211		.WORD	013211
2668	005152	105504		.WORD	105504
2669	005154	042642		.WORD	042642
2670	005156	021321		.WORD	021321
2671	005160	110550		.WORD	110550
2672	005162	044264		.WORD	044264
2673	005164	022132		.WORD	022132
2674	005166	011055		.WORD	011055
2675	005170	104426		.WORD	104426
2676	005172	042213		.WORD	042213
2677					
2678	005174	177777	PAT7:	.WORD	177777
2679					
2680	005176	045513	PAT8:	.WORD	045513
2681	005200	122645		.WORD	122645
2682	005202	151322		.WORD	151322
2683	005204	064551		.WORD	064551
2684	005206	132264		.WORD	132264
2685	005210	055132		.WORD	055132
2686	005212	026455		.WORD	026455
2687	005214	113226		.WORD	113226
2688	005216	045513		.WORD	045513
2689	005220	122645		.WORD	122645
2690	005222	151322		.WORD	151322
2691	005224	064551		.WORD	064551

GLOBAL DATA AND CONSTANTS

2692	005226	132264				.WORD	132264
2693	005230	055132				.WORD	055132
2694	005232	026455				.WORD	026455
2695	005234	113226				.WORD	113226
2696							
2697	005236	125252			PAT9:	.WORD	125252
2698							
2699	005240	155555			PAT10:	.WORD	155555
2700							
2701	005242				ENDMOD		
2702							
2703					.SBTTL	GLOBAL MESSAGES	
2704							
2708	005242				BGNMOD	GLBTXT	
2709	005242	123	105	105	MSEEK:	.ASCIZ	/SEEK /
2710	005250	122	104	040	MREAD:	.ASCIZ	/RD DATA /
2711	005261	122	104	040	MREADH:	.ASCIZ	/RD HDR /
2712	005271	127	122	124	MWRCHK:	.ASCIZ	/WRT CHECK/
2713	005303	127	122	124	MWRITE:	.ASCIZ	/WRT DATA /
2714	005315	107	105	124	MGTSTA:	.ASCIZ	/GET STAT /
2715	005327	127	111	124	MDATCP:	.ASCIZ	/WITH DATA CMP /
2716	005346	127	111	124	MHDRCP:	.ASCIZ	/WITH HDR CMP /
2717	005364	106	117	122	M40HDR:	.ASCIZ	/FOR 40 HDRS/
2718	005400	127	111	124	MWRSET:	.ASCIZ	/WITH RESET /
2719	005414	117	120	105	MOPER:	.ASCIZ	/OPER: /
2720	005423	122	105	123	MRESLT:	.ASCIZ	/RESULT: /
2721	005434	125	116	114	MLOAD:	.ASCIZ	/UNLD DRV/
2722	005445	114	104	040	MCYLUP:	.ASCIZ	/LD DRV /
2723	005455	106	117	114	MOUTIN:	.ASCIZ	/FOL 0 TO CC SEEK/
2724	005476	106	117	114	MINOUT:	.ASCIZ	/FOL 255 TO CC SEEK/
2725	005521	106	117	114	MFOLWRT:	.ASCIZ	/FOL WRT (NO SEEK)/
2726	005543	101	104	112	MREVSK:	.ASCIZ	/ADJ CYL WRTTN AFTER REV SK/
2727	005576	101	104	112	MFWSK:	.ASCIZ	/ADJ CYL WRTTN AFTER FWD SK/
2728	005631	123	113	040	MFWSKO:	.ASCIZ	/SK FWD,WRT - SK REV,OVERWRT/
2729	005665	123	113	040	MRESKO:	.ASCIZ	/SK REV,WRT - SK FWD,OVERWRT/
2730	005721	117	116	040	MBADAD:	.ASCIZ	/ON BAD SEC FILES/
2731	005742	103	101	116	MBADSF:	.ASCIZ	/CANNOT GET BAD SEC FILES/
2732	005773	102	101	104	MFHTER:	.ASCIZ	/BAD SEC FILE FMT ERR/
2733	006020	124	117	117	MTHBS:	.ASCIZ	/TOO MANY BAD SEC /
2734	006042	102	125	123	BASADD:	.ASCIZ	/BUS ADD=/
2735	006053	104	122	126	DRVNAM:	.ASCIZ	/DRV=/
2736	006060	104	122	126	NOPIR:	.ASCIZ	/DRV DID NOT REC'R FROM PWR FAIL/
2737	006120	122	114	103	CSNAM:	.ASCIZ	/RLCS/
2738	006125	122	114	102	BANAM:	.ASCIZ	/RLBA/
2739	006132	122	114	104	DANAM:	.ASCIZ	/RLDA/
2740	006137	122	114	115	MPNAM:	.ASCIZ	/RLMP/
2741	006144	117	120	040	LAB1:	.ASCIZ	/OP INIT = /
2742	006157	117	120	040	LAB2:	.ASCIZ	/OP DONE = /
2743	006172	127	117	122	MWORD:	.ASCIZ	/WORD /
2744	006200	111	116	124	MTOLOW:	.ASCIZ	/INTRPT TOO LATE/
2745	006220	116	117	040	MDRRES:	.ASCIZ	/NO DRV RESPONSE/
2746	006240	116	117	040	MNOINT:	.ASCIZ	/NO INTRPT ON CMND COMPLETE/
2747	006273	103	116	124	MCONHG:	.ASCIZ	/CNTLR HUNG /
2748	006307	105	122	122	MNOCLR:	.ASCIZ	/ERR DID NOT CLR/
2749	006327	126	117	114	VCNRS:	.ASCIZ	/VOL CHK NOT RSET/
2750	006350	125	116	130	UNXERR:	.ASCIZ	/UNXPCTD ERR/
2751	006365	040	124	105	TSTLAB:	.ASCIZ	/ TEST/

GLOBAL MESSAGES

2753	006373	115	101	116	MISTST: .ASCIZ	/MAN INTERVENT STAT/
2754	006416	123	124	101	NSTACHG: .ASCIZ	/STATE CHG/
2755	006430	123	120	116	SPDERR: .ASCIZ	/SPNDL TIMEOUT FAILED TO SET/
2756	006464	106	101	111	GSTER1: .ASCIZ	/FAIL FORCING DRV SEL ERR/
2757	006515	111	116	111	INITST: .ASCIZ	/INIT STATE/
2758	006530	104	122	126	T05ERR: .ASCIZ	/DRV SELECT/
2759	006543	104	122	126	T09ERR: .ASCIZ	/DRV RDY/
2760	006553	123	105	105	T10ERR: .ASCIZ	/SEEK SGN SWITCH/
2761	006573	110	104	040	T12ERR: .ASCIZ	/HD SWITCH/
2762	006605	122	104	040	T13ERR: .ASCIZ	/RD HDR (P1)/
2763	006621	122	104	040	T14ERR: .ASCIZ	/RD HDR (P2)/
2764	006635	127	122	124	T16ERR: .ASCIZ	/WRT LCK/
2765	006645				P2T01E:	
2766	006645	104	111	106	P2T02E: .ASCIZ	/DIFF OF 1 SEEK/
2767	006664	124	105	123	NOTST: .ASCIZ	/TEST CANNOT BE PERFORMED...NO P CLOCK OR SOFTWARE CLOCK/
2768	006754	104	122	126	NOCTLR: .ASCIZ	/DRV DROPPED - NO CNTLR/
2769	007003	104	122	126	NOTRDY: .ASCIZ	/DRV DROPPED - NOT RDY/
2804	007031	110	104	123	HDMOVF: .ASCIZ	/HDS FAILED TO MOVE IN 10 TRIES/
2806	007070	103	131	114	CYLPER: .ASCIZ	/CYL PORTION OF HDRS DIFFER WHEN READ FROM TRK 0 & 1/
2807	007154	110	105	101	HAMES1: .ASCIZ	/HEAD ALIGN. RSET WRT LCK TO SEL HD 0, SET FOR HD 1/
2808	007237	124	131	120	HAMES2: .ASCIZ	&TYPE "CTL/C" TO GET BACK TO SUPVR CMD MODE AND THEN TYPE "CONT" &
2809	007343	111	106	040	HAMES3: .ASCIZ	/IF HD SEL TP (21, 22) DO NOT EXIST/
2810	007406	107	116	104	HAMES4: .ASCIZ	/GND NULL DET ON DRV LGC MOD DISABLE SEEK TIME OUT/
2811	007470	101	102	117	OPR002: .ASCIZ	/ABOVE CONDITIONS MET/
2812	007515	127	101	123	OPR003: .ASCIZ	/WAS LOAD DEPRESSED/
2813	007540	103	110	113	OPR1: .ASCIZ	/CHK DRV IS UNLDED, COVER OPN, AND WRTE LCKED /
2814	007616	103	114	117	OPR2: .ASCIZ	/CLOSE COVER & RST WRT LCK /
2815	007651	120	122	105	OPR3: .ASCIZ	/PRESS LOAD /
2816	007665	120	122	105	OPR6: .ASCIZ	/PRESS LOAD & WAIT FOR RDY /
2817	007720	122	105	115	OPR7: .ASCIZ	/REMOVE ADR PLGS EXCPT /
2818	007747	111	116	123	OPR8: .ASCIZ	/INSRT ADR PLG /
2819	007766	111	116	040	OPR9: .ASCIZ	/IN ALL DRVS /
2820	010003	111	116	123	OPR10: .ASCIZ	/INSUFFICIENT DRVS FOR DRV SEL ERR TST/
2821	010051	122	120	114	OPR11: .ASCIZ	/RPLCE ADR PLGS AS BEFORE/
2823	010102	122	105	123	OPR12: .ASCIZ	/RESET WRT LCK /
2824	010121	123	105	124	OPR12A: .ASCIZ	/SET WRT LCK/
2825	010135	117	116	040	OPR1A: .ASCIZ	/ON /
2826	010141	117	116	040	OPR1B: .ASCIZ	/ON DRV /
2827	010151	125	116	104	UNDTST: .ASCIZ	/UNDER TEST/
2828	010164	123	105	124	OPR004: .ASCIZ	/SET WRT LCK /
2829	010201	104	111	106	DIFWD: .ASCIZ	/DIFF /
2830	010207	123	107	116	SGNMD: .ASCIZ	/SGN /
2831	010214	110	104	040	HDMD: .ASCIZ	/HD /
2832	010220	123	105	103	SECMD: .ASCIZ	/SEC /
2833	010225	103	131	114	CYLMD: .ASCIZ	/CYL /
2834	010232	106	122	117	FRMD: .ASCIZ	/FROM /
2835	010240	040	102	131	BYPSSM: .ASCIZ	/ BYPASSED /
2836	010253	122	117	125	SEQMES: .ASCIZ	/ROUTINE TRACE SEQ:/
2837	010276	104	122	126	STATES: .ASCIZ	/DRV STAT/
2838	010307	102	101	104	BSNSTR: .ASCIZ	/BAD SEC FILES NOT STRD. ALL SEC ASSUMED OK./
2839	010363	124	117	124	TCERR: .ASCIZ	/TOTAL CMP ERRS: /
2840						
2841						
2842	010404	104	122	126	MDRDY: .ASCIZ	/DRV RDY /
2843	010415	103	117	116	MCERR: .ASCIZ	/CONT ERR /
2844	010427	110	104	122	MHCRC: .ASCIZ	/HDR CRC/
2845	010437	104	101	124	MDCRC: .ASCIZ	/DATA CRC/

GLOBAL MESSAGES

2846	010450	110	104	122	MHNF: .ASCIZ /HDR NOT FND/
2847	010464	104	101	124	MDLT: .ASCIZ /DATA LATE/
2848	010476	110	104	122	MHFCRC: .ASCIZ &HDR NOT FND/HDR CRC/OPI&
2849	010526	104	122	126	MDRERR: .ASCIZ /DRV ERR /
2851	010537	123	105	114	MHSTA: .ASCIZ /SEL'D HD /
2852	010551	126	117	114	MVOLCK: .ASCIZ /VOL CHK /
2853	010562	103	117	126	MCOSTA: .ASCIZ /COVER OPEN/
2854	010575	102	122	125	MHSTA: .ASCIZ /BRUSH HOME/
2855	010610	127	122	124	MMLSTA: .ASCIZ /WRT LCK /
2856	010621	110	104	123	MHOSTA: .ASCIZ /HDS OUT /
2858	010632	104	122	126	MDSERR: .ASCIZ /DRV SEL ERR /
2859	010647	104	122	126	MDRVST: .ASCIZ /DRV STATE /
2860	010662	123	120	111	MSPERR: .ASCIZ /SPIN TIMEOUT /
2861	010700	127	122	124	MHGERR: .ASCIZ /WRT GAT ERR /
2862	010715	123	105	105	MSTERR: .ASCIZ /SEEK TIMEOUT /
2863	010733	110	105	101	MHCERR: .ASCIZ /HEAD CUR ERR /
2864	010751	127	122	124	MHDERR: .ASCIZ /WRT DAT ERR /
2865	010766	117	120	122	MOPERR: .ASCIZ /OPR-INC/
2866	010776	110	104	122	MHDERR: .ASCIZ &HDR/DAT ERR &
2867	011013	110	104	122	MFLERR: .ASCIZ &HDR NOT FND/DAT LATE &
2868	011041	116	055	130	MNEERR: .ASCIZ /N-X-MEM /
2869	011052	103	131	114	MCYLOC: .ASCIZ /CYL /
2870	011057	103	101	116	MNDRST: .ASCIZ /CANNOT GET DRV STAT/
2871	011103	125	116	113	MUNDEF: .ASCIZ /UNKN DRV STATE-NO RDY,NO ERR,HDS OUT/
2872	011150	106	101	111	MRLFAL: .ASCIZ /FAIL TO RELD HDS AFTER ERR CLEAR/
2873	011211	127	122	124	MWRTAB: .ASCIZ /WRT ABORTED/
2874	011225	040	117	126	MEXERS: .ASCIZ / OVER ERR LIMIT - UNIT DROPPED /
2875	011265	040	105	122	MERRS: .ASCIZ / ERROR/
2876	011274	207	377	377	BELL: .ASCIZ <207><377><377>
2877					
2878					; RESULT SETTINGS
2879	011300	111	123	040	RESE3: .ASCIZ /IS /
2880	011304	040	123	102	RESE4: .ASCIZ / SB /
2881					
2882					; RESULT CONDITIONS
2883	011311	040	111	116	RESE5: .ASCIZ / IN /
2884	011316	040	117	106	RESE6: .ASCIZ / OF /
2885	011323	123	124	101	STATE2: .ASCIZ /STATE 2/
2886	011333	123	124	101	STATE3: .ASCIZ /STATE 3/
2887	011343	123	124	101	STATE5: .ASCIZ /STATE 5/
2889	011353	123	105	105	CDRDY: .ASCIZ &SEEK W/O MOTIONE&
2891	011373	061	123	124	C10MS: .ASCIZ /1ST 3 MS/
2892	011404	065	060	060	C500MS: .ASCIZ /500MS/
2893	011412	103	131	103	CCYLUP: .ASCIZ /CYCLE UP/
2894	011423	104	101	124	CAFDT: .ASCIZ /DATA XFR/
2895	011434	065	040	123	C5SEC: .ASCIZ /5 SEC/
2896					
2897	011442	045	116	045	FMTOP1: .ASCIZ /#N#T#N#T#T#06#S#T#01#N/
2898	011471	045	116	045	FMTOP2: .ASCIZ /#N#T#01#S1#T#01#N/
2899	011513	045	116	045	FMTOP3: .ASCIZ /#N#T#01#S1#T#T#N/
2900	011534	045	124	045	FMT1: .ASCIZ /#T#T/
2901	011541	045	116	045	FMT1.1: .ASCIZ /#N#T#T/
2902	011550	045	124	000	FMT2: .ASCIZ /#T/
2903	011553	045	116	000	FMT3: .ASCIZ /#N/
2904	011556	045	116	045	FMT4: .ASCIZ /#N#T#T#N/
2905	011567	045	116	045	FMT5: .ASCIZ /#N#T#06#S1#T#01/
2906	011607	045	116	045	FMT6: .ASCIZ /#N#S11#T#S4#T#S4#T#S4#T#S4#T#S2#T/

GLOBAL MESSAGES

```

2907 011651 045 116 045 FMT7: .ASCIZ /#N#T#06#S2#06#S2#06#S2#06#S3#03#S2#01#N/
2908 011721 045 116 045 FMT8: .ASCIZ /#N#T#06#S2#06#S2#06#S2#06/
2909 011753 045 116 045 FMT9: .ASCIZ /#N#T/
2910 011760 045 124 045 FMT11: .ASCIZ /#T#01/
2911 011766 045 124 045 FMT12: .ASCIZ /#T#03/
2912 011774 045 116 045 FMT13: .ASCIZ /#N#S11#T#03#S1#T#03#S1#T#01#S1#T#01/
2913 012040 045 116 045 FMT14: .ASCIZ /#N#T#T#D3#S1#T#06#S1#T#06/
2914 012072 045 116 045 FMT15: .ASCIZ /#N#S11#T#D3#S1#T#06#S1#T#06/
2915 012126 045 116 045 FMT16: .ASCIZ /#N#S5#06/
2916 012137 045 123 061 FMT17: .ASCIZ /#S10#T#N#S11#06#N/
2917 012161 045 116 045 FMT18: .ASCIZ /#N#S15#T#S5#T#S4#T#S5#T#N/
2918 012213 045 124 045 FMT19: .ASCIZ /#T#S4#D6#S4#D6#S4#D6#S4#D6#N/
2919 012250 045 124 045 FMT20: .ASCIZ /#T#S2#D6#S14#D6#S4#D6#N/
2920 012300 045 124 045 FMT21: .ASCIZ /#T#S12#D6#S14#D6#N/
2921 012323 045 116 045 FMT22: .ASCIZ /#N#S11#T#03#S1#T#01#S1#T#02/
2922 012357 045 124 045 FMT23: .ASCIZ /#T#T#T#01#N/
2923 012373 045 116 045 FMT24: .ASCIZ /#N#T/
2924 012400 045 116 045 FMT25: .ASCIZ /#N#D2#T/
2925 012410 045 116 045 FMT26: .ASCIZ /#N#S1#T#D4#T#T#D3#N/
2926 012434 045 116 045 FMT27: .ASCIZ /#N#T#D3#T#D3#N/
2927 012453 045 116 045 FMT28: .ASCIZ /#N#T#T#T/
2928 012464 ENDMOD

```

.SBTTL ERROR MESSAGES

```

2933
2934
2935
2936 012464 BGNMOD GLBERR
2937 ; ERR1 R3 POINTS TO RESULT MESSAGE
2938 ; RESULT: (R3)
2939
2940 ; ERR2 R3 POINTS TO RESULT NAME
2941 ; RESULT: (R3) IS 1 SB 0
2942
2943 ; ERR3 R3 POINTS TO RESULT NAME
2944 ; RESULT: (R3) IS 0 SB 1
2945
2946 ; ERR4 R3 POINTS TO RESULT NAME
2947 ; R4 POINTS TO RESULT CONDITIONS
2948 ; RESULT: (R3) IS 1 SB 0 (R4)
2949
2950 ; ERR5 R3 POINTS TO RESULT NAME
2951 ; R4 POINTS TO RESULT CONDITIONS
2952 ; RESULT: (R3) IS 0 SB 1 (R4)
2953
2954 ; ERR6 RESULT ROUTINE DETERMINES WHICH ERROR(S) ARE SET AND
2955 ; REPORTS ALL
2956 ; RESULT: "ERROR" IS 1 SB 0
2957
2958 ; ERR7 DRIVE STATE ERROR REPORT
2959 ; R3 CONTAINS EXPECTED STATE
2960 ; T.STAT CONTAINS BAD STATE
2961 ; RESULT: DRIVE STATE IS (T.STAT) SB (R3)
2962
2963 ; ERR8 HEAD POSITIONING ERROR REPORT
2964 ; NEWCYL CONTAINS EXPECTED CYLINDER
2965 ; HDWRD1 CONTAINS BAD CYLINDER
2966 ; RESULT: CYLINDER IS (HDWRD1) SB (NEWCYL)
2967

```

ERROR MESSAGES

```

2968      ;      ERR9  UTILITY RESULT REPORT
2969      ;      R3 POINTS TO RESULT NAME
2970      ;      R4 POINTS TO VALUE 1
2971      ;      R5 POINTS TO VALUE 2
2972      ;      RESULT: (R3-NAME) IS (R4-VALUE 1) SB (R5-VALUE 2)
2973
2974      ;      ERR10 COMPARE ERROR REPORT
2975      ;      R3 CONTAINS THE BAD WORD NUMBER
2976      ;      R4 POINTS TO BAD WORD
2977      ;      R5 POINTS TO GOOD WORD
2978      ;      RESULT: WORD (R3) IS (R4) SB (R5)
2979
2980      .NLIST MD,ME
2981
2982      BGNMSG  ERR1
2983      012464 105767 170675      TSTB      NOERCT      ;TEST IF ERROR COUNTING INHIBITED
2984      012470 001002      BNE      1$      ;YES - SKIP
2985      012472 005277 170460      INC      @ERRPOINT ;ELSE BUMP ERROR COUNT
2986      012476 010146      1$:      MOV      R1,-(SP) ;STORE R1
2987      012500 004767 011106      JSR      PC,RPTOP ;REPORT OPERATION
2988      012504 012721 000001      MOV      #1,(R1)+ ;SET PARAM NUMBER
2989      012510 010321      MOV      R3,(R1)+ ;INSERT MESSAGE ADDRESS POINTER
2990      012512 004767 011662      JSR      PC,RPTRES ;REPORT RESULTS
2991      012516 004767 012064      JSR      PC,RPTREM ;REPORT REMAINDER
2992      012522 012601      MOV      (SP)+,R1 ;RESTORE R1
2993      012524 004767 003714      JSR      PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
2994      012530      ENDMSG
2995      012530      L10000:
2996      012530 104423      TRAP      C#MSG
2997
2998      BGNMSG  ERR2
2999      012532 005277 170420      INC      @ERRPOINT ;BUMP ERROR COUNT
3000      012536 010146      MOV      R1,-(SP) ;STORE R1
3001      012540 004767 011046      JSR      PC,RPTOP ;REPORT OPERATION
3002      012544 012721 000003      MOV      #3,(R1)+ ;SET PARAM NUMBER
3003      012550 010321      MOV      R3,(R1)+ ;INSERT NAME ADD POINTER
3004      012552 012721 000001      MOV      #1,(R1)+ ;SET IS VALUE
3005      012556 005021      CLR      (R1)+ ;SET SB VALUE
3006      012560 004767 011614      JSR      PC,RPTRES ;REPORT RESULTS
3007      012564 004767 012016      JSR      PC,RPTREM ;REPORT REMAINDER
3008      012570 012601      MOV      (SP)+,R1 ;RESTORE R1
3009      012572 004767 003646      JSR      PC,CKERLM ;GO CHECK IF ERROR COUNT EXCEEDED
3010      012576      ENDMSG
3011      012576      L10001:
3012      012576 104423      TRAP      C#MSG
3013
3014      BGNMSG  ERR3
3015      012600 005277 170352      INC      @ERRPOINT ;BUMP ERROR COUNT
3016      012604 010146      MOV      R1,-(SP) ;STORE R1
3017      012606 004767 011000      JSR      PC,RPTOP ;REPORT OPERATION
3018      012612 012721 000003      MOV      #3,(R1)+ ;SET PARAM NUMBER
3019      012616 010321      MOV      R3,(R1)+ ;INSERT NAME ADD POINTER
3020      012620 005021      CLR      (R1)+ ;SET IS VALUE
3021      012622 012721 000001      MOV      #1,(R1)+ ;SET SB VALUE
3022      012626 004767 011546      JSR      PC,RPTRES ;REPORT RESULTS
3023      012632 004767 011750      JSR      PC,RPTREM ;REPORT REMAINDER
3024      012636 012601      MOV      (SP)+,R1 ;RESTORE R1

```

ERROR MESSAGES

```

3021 012640 004767 003600          JSR    PC,CKERLM          ;GO CHECK IF ERROR COUNT EXCEEDED
3022 012644          ENDMSG
      012644          L10002:
      012644 104423          TRAP   C#MSG

3023
3024 012646          BGNMSG  ERR4
3025 012646 005277 170304          INC    @ERRPOINT          ;BUMP ERROR COUNT
3026 012652 010146          MOV    R1,-(SP)           ;STORE R1
3027 012654 004767 010732          JSR    PC,RPTOP           ;REPORT OPERATION
3028 012660 012721 000004          MOV    #4,(R1)+           ;SET PARAM NUMBER
3029 012664 010321          MOV    R3,(R1)+           ;INSERT NAME ADD POINTER
3030 012666 012721 000001          MOV    #1,(R1)+           ;SET IS VALUE
3031 012672 005021          CLR    (R1)+              ;SET SB VALUE
3032 012674 010411          MOV    R4,(R1)            ;INSERT ADD OF CONDITION POINTER
3033 012676 004767 011476          JSR    PC,RPTRES          ;REPORT RESULTS
3034 012702 004767 011700          JSR    PC,RPTREM          ;REPORT REMAINDER
3035 012706 012601          MOV    (SP)+,R1           ;RESTORE R1
3036 012710 004767 003530          JSR    PC,CKERLM          ;GO CHECK IF ERROR COUNT EXCEEDED
3037 012714          ENDMSG
      012714          L10003:
      012714 104423          TRAP   C#MSG

3038
3039 012716          BGNMSG  ERR5
3040 012716 005277 170234          INC    @ERRPOINT          ;BUMP ERROR COUNT
3041 012722 010146          MOV    R1,-(SP)           ;STORE R1
3042 012724 004767 010662          JSR    PC,RPTOP           ;REPORT OPERATION
3043 012730 012721 000004          MOV    #4,(R1)+           ;SET PARAM NUMBER
3044 012734 010321          MOV    R3,(R1)+           ;INSERT NAME ADD POINTER
3045 012736 005021          CLR    (R1)+              ;SET IS VALUE
3046 012740 012721 000001          MOV    #1,(R1)+           ;SET SB VALUE
3047 012744 010411          MOV    R4,(R1)            ;INSERT ADD OF CONDITION POINTER
3048 012746 004767 011426          JSR    PC,RPTRES          ;REPORT RESULTS
3049 012752 004767 011630          JSR    PC,RPTREM          ;REPORT REMAINDER
3050 012756 012601          MOV    (SP)+,R1           ;RESTORE R1
3051 012760 004767 003460          JSR    PC,CKERLM          ;GO CHECK IF ERROR COUNT EXCEEDED
3052 012764          ENDMSG
      012764          L10004:
      012764 104423          TRAP   C#MSG

3053
3054 012766          BGNMSG  ERR6
3055 012766 105767 170373          TSTB   NOERCT              ;TEST IF ERROR COUNTING INHIBITED
3056 012772 001002          BNE    17#                 ;YES - SKIP
3057 012774 005277 170156          INC    @ERRPOINT          ;ELSE BUMP ERROR COUNT
3058 013000 010146          17#:  MOV    R1,-(SP)           ;STORE R1
3059 013002 010346          MOV    R3,-(SP)           ;STORE R3
3060 013004 010446          MOV    R4,-(SP)           ;STORE R4
3061 013006 010546          MOV    R5,-(SP)           ;STORE R5
3062 013010 004767 010576          JSR    PC,RPTOP           ;REPORT OPERATION
3063 013014 012721 000003          MOV    #3,(R1)+           ;SET PARAM NUMBER
3064 013020 012761 000001 000002  MOV    #1,2(R1)           ;INSERT IS VALUE
3065 013026 005067 170072          CLR    TEMP3              ;CLEAR FOR STATUS STORAGE
3066 013032 016703 170006          MOV    T,CS,R3            ;GET T.CS
3067 013036 042703 177761          BIC    #177761,R3         ;AND CLEAR ALL BUT FUNCTION
3068 013042 022703 000004          CMP    #4,R3              ;CHECK IF IT WAS GET STATUS
3069 013046 001443          BEQ    1#                 ;YES - STATUS IS IN T.MP, SKIP
3070 013050 012762 000003 000004  MOV    #GETSTAT,RLDA(R2) ;ELSE DO GET STATUS
3071 013056 012703 000004          MOV    #4,R3

```

ERROR MESSAGES

3072	013062	056703	167744		BIS	RLDRV,R3		
3073	013066	010362	000000		MOV	R3,RLCS(R2)		
3074					WAITUS	10.	;WAIT FOR CONTROLLER READY	;JSD REV A
3075	013072				WAITUS	50.	;WAIT FOR CONTROLLER READY	;JSD REV A
3076	013122	032762	000200	000000	BIT	@CRDYMSK,RLCS(R2)	;TEST IF READY	
3077	013130	001003			BNE	10#	;YES - SKIP	
3078	013132	012703	001000	9#:	MOV	@BIT9,R3	;ELSE SET NO DRIVE STATUS BIT	
3079	013136	000413			BR	2#	;IN MESSAGE WORD AND SKIP	
3080	013140	016203	000006	10#:	MOV	RLMP(R2),R3	;STORE STATUS FOR REPORT	
3081	013144	010367	167754		MOV	R3,TEMP3		
3082	013150	116703	167751		MOVB	TEMP3+1,R3	;GET ERROR BITS IN PROPER POSITION	
3083	013154	000402			BR	13#		
3084	013156	116703	167671	1#:	MOVB	T.MP+1,R3	;GET ERROR BITS FROM MP REG	
3085	013162	042703	177442	13#:	BIC	@177442,R3	;CLEAR UNUSED BITS	
3086	013166	016704	167652	2#:	MOV	T.CS,R4	;GET ERROR BITS FROM CS REG	
3087	013172	042704	001777		BIC	@1777,R4	;CLEAR UNUSED BITS	
3088	013176	050403			BIS	R4,R3	;MAKE ONE WORD OF POSSIBLE ERRORS	
3089	013200	032703	002000		BIT	@OPIERR,R3	;TEST IF OPI SET	
3090	013204	001442			BEQ	115#	;NO - SKIP	
3091	013206	032703	010000		BIT	@HNFERR,R3	;TEST IF HDR NOT FOUND ERROR	
3092	013212	001026			BNE	107#	;YES - SKIP	
3093	013214	032703	004000		BIT	@HCRCERR,R3	;TEST IF HDR CRC ERR	
3094	013220	001020			BNE	105#	;YES - SKIP	
3095	013222	012704	010766		MOV	@MOPERR,R4	;SET OPI ALONE MESSAGE	
3096	013226			100#:	PRINTB	@FMT28,@MRSLT,R4,@MERRS	;REPORT ERROR	
	013226	012746	011265		MOV	@MERRS,-(SP)		
	013232	010446			MOV	R4,-(SP)		
	013234	012746	005423		MOV	@MRSLT,-(SP)		
	013240	012746	012453		MOV	@FMT28,-(SP)		
	013244	012746	000004		MOV	@4,-(SP)		
	013250	010600			MOV	SP,R0		
	013252	104414			TRAP	C#PNTB		
	013254	062706	000012		ADD	@12,SP		
3097	013260	000430			BR	120#	;SKIP	
3098	013262	012704	010427	105#:	MOV	@HCRC,R4	;HDR CRC MESSAGE	
3099	013266	000757			BR	100#		
3100	013270	032703	004000	107#:	BIT	@HCRCERR,R3	;TEST IF HCRC WITH HDR NOT FND	
3101	013274	001003			BNE	109#	;YES - SKIP	
3102	013276	012704	010450		MOV	@HNF,R4	;MESSAGE HEADER NOT FOUND	
3103	013302	000751			BR	100#		
3104	013304	012704	010476	109#:	MOV	@HNFRC,R4	;HNF AND HCRC MESSAGE	
3105	013310	000746			BR	100#	;SKIP	
3106	013312	032703	004000	115#:	BIT	@DCKERR,R3	;TEST IF DATA CHECK SET, NOT OPI	
3107	013316	001403			BEQ	118#	;NO - SKIP	
3108	013320	012704	010437		MOV	@MDCRC,R4	;SET MESSAGE DATA CHECK	
3109	013324	000740			BR	100#	;SKIP	
3110	013326	032703	010000	118#:	BIT	@DLTERR,R3	;TEST IF DATA LATE ERROR	
3111	013332	001403			BEQ	120#	;NO - SKIP	
3112	013334	012704	010464		MOV	@MDLT,R4	;SET MESSAGE DATA LATE	
3113	013340	000732			BR	100#	;SKIP	
3114	013342	012705	100000	120#:	MOV	@BIT15,R5	;SET BIT POINTER FOR TEST	
3115	013346	005004			CLR	R4	;CLEAR R4 FOR TABLE COUNT	
3116	013350	030503		3#:	BIT	R5,R3	;TEST IF BIT IS SET	
3117	013352	001005			BNE	6#	;YES - SKIP TO REPORT	
3118	013354	005724		4#:	TST	(R4)+	;ELSE BUMP TABLE POINTER	
3119	013356	000241			CLC		;CLEAR CARRY	
3120	013360	006005			ROR	R5	;SHIFT BIT POINTER TO NEXT BIT	

ERROR MESSAGES

3121	013362	001372				BNE	3#		;LOOP IF NOT 0
3122	013364	000405				BR	7#		;ELSE REPORT REMAINDER
3123	013366	016411	002320		6#:	MOV	RESTBL(R4),(R1)		;INSERT NAME ADDRESS
3124	013372	004767	011002			JSR	PC,RPTRES		;REPORT RESULTS
3125	013376	000766				BR	4#		;GET NEXT BIT
3126	013400	004767	011202		7#:	JSR	PC,RPTREM		;REPORT REMAINDER
3127	013404	005767	167514			TST	TEMP3		;TEST IF ANY NEW STATUS
3128	013410	001414				BEQ	15#		;NO - SKIP
3129	013412					PRINTB	#FMT17,#STAMES,TEMP3		
	013412	016746	167506			MOV	TEMP3,-(SP)		
	013416	012746	010276			MOV	#STAMES,-(SP)		
	013422	012746	012137			MOV	#FMT17,-(SP)		
	013426	012746	000003			MOV	#3,-(SP)		
	013432	010600				MOV	SP,R0		
	013434	104414				TRAP	C#PNTB		
	013436	062706	000010			ADD	#10,SP		
3130	013442	032767	004000	167374	15#:	BIT	#DCKERR,T.CS		;TEST IF DATA CHECK ERROR
3131	013450	001453				BEQ	25#		;NO - SKIP
3132	013452	032767	002000	167364		BIT	#OPIERR,T.CS		;TEST IF OPI SET
3133	013460	001047				BNE	25#		;YES - SKIP
3134	013462	005067	167326			CLR	MORECE		;CLEAR COMPARE ERROR COUNT
3135	013466	012701	000200			MOV	#128.,R1		;SET COMPARE LENGTH
3136	013472	012703	000001			MOV	#1,R3		;SET WORD COUNT
3137	013476	012705	004364			MOV	#OBUF,R5		;SET GOOD WORD POINTER
3138	013502	012704	003764			MOV	#IBUFF,R4		;SET TEST WORD POINTER
3139	013506	021514			18#:	CMP	(R5),(R4)		;CHECK WORD
3140	013510	001427				BEQ	19#		;GOOD - SKIP
3141	013512	026727	167276	000012		CMP	MORECE,#10.		;TEST IF COMPARE LIMIT REACHED
3142	013520	003021				BGT	20#		;YES - SKIP
3143	013522					PRINTB	#FMT15,#MWORD,R3,#RESE3,(R4),#RESE4,(R5)		
	013522	011546				MOV	(R5),-(SP)		
	013524	012746	011304			MOV	#RESE4,-(SP)		
	013530	011446				MOV	(R4),-(SP)		
	013532	012746	011300			MOV	#RESE3,-(SP)		
	013536	010346				MOV	R3,-(SP)		
	013540	012746	006172			MOV	#MWORD,-(SP)		
	013544	012746	012072			MOV	#FMT15,-(SP)		
	013550	012746	000007			MOV	#7,-(SP)		
	013554	010600				MOV	SP,R0		
	013556	104414				TRAP	C#PNTB		
	013560	062706	000020			ADD	#20,SP		
3144	013564	005267	167224		20#:	INC	MORECE		;BUMP ERROR COUNTER
3145	013570	022524			19#:	CMP	(R5),.(R4).		;BUMP POINTERS
3146	013572	005203				INC	R3		;BUMP COUNTER
3147	013574	005301				DEC	R1		;DEC LENGTH COUNT
3148	013576	001343				BNE	18#		;LOOP IF NOT DONE
3149	013600	005767	167210		25#:	TST	MORECE		;TEST IF ANY COMPARE ERRORS
3150	013604	001421				BEQ	27#		;NO - SKIP
3151	013606	012701	000200			MOV	#128.,R1		;SET COMPARE LENGTH
3152	013612					PRINTB	#FMT27,#TCERR,MORECE,#RESE6,R1		
	013612	010146				MOV	R1,-(SP)		
	013614	012746	011316			MOV	#RESE6,-(SP)		
	013620	016746	167170			MOV	MORECE,-(SP)		
	013624	012746	010363			MOV	#TCERR,-(SP)		
	013630	012746	012434			MOV	#FMT27,-(SP)		
	013634	012746	000005			MOV	#5,-(SP)		
	013640	010600				MOV	SP,R0		

ERROR MESSAGES

	013642	104414			TRAP	C#PNTB	
	013644	062706	000014		ADD	#14,SP	
3153	013650	012605		27#:	MOV	(SP)+,R5	;RESTORE R5, 4, 3, 1
3154	013652	012604			MOV	(SP)+,R4	
3155	013654	012603			MOV	(SP)+,R3	
3156	013656	012601			MOV	(SP)+,R1	
3157	013660	004767	002560		JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
3158	013664			ENDMSG			
	013664			L10005:			
	013664	104423			TRAP	C#MSG	
3159				BGNMSG	ERR7		
3160	013666				INC	@ERRPOINT	;BUMP ERROR COUNT
3161	013666	005277	167264		MOV	R1,-(SP)	;STORE R1
3162	013672	010146			JSR	PC,RPTOP	;REPORT OPERATION
3163	013674	004767	007712		MOV	#3,(R1)+	;SET PARAM NUMBER
3164	013700	012721	000003		MOV	@MDRVST,(R1)+	;INSERT NAME ADD POINTER
3165	013704	012721	010647		MOV	T,STAT,(R1)+	;INSERT IS VALUE
3166	013710	016721	167144		MOV	R3,(R1)	;INSERT SB VALUE
3167	013714	010311			JSR	PC,RPTRES	;REPORT RESULTS
3168	013716	004767	010456		JSR	PC,RPTREM	;REPORT REMAINDER
3169	013722	004767	010660		MOV	(SP)+,R1	;RESTORE R1
3170	013726	012601			JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
3171	013730	004767	002510	ENDMSG			
3172	013734			L10006:			
	013734				TRAP	C#MSG	
	013734	104423		BGNMSG	ERR8		
3173					INC	@ERRPOINT	;BUMP ERROR COUNT
3174	013736				MOV	R1,-(SP)	;STORE R1
3175	013736	005277	167214		MOV	R3,-(SP)	;STORE R3
3176	013742	010146			JSR	PC,RPTOP	;REPORT OPERATION
3177	013744	010346			MOV	#3,(R1)+	;SET PARAM NUMBER
3178	013746	004767	007640		MOV	@MCYLOC,(R1)+	;INSERT NAME ADD POINTER
3179	013752	012721	000003		MOV	HDWRD1,(R1)	;GET HEADER WORD
3180	013756	012721	011052		MOV	#7,R3	;SET SHIFT COUNT
3181	013762	016711	167064		CLC		
3182	013766	012703	000007	3#:	ROR	(R1)	;ALIGN CHAR FOR PRINTING
3183	013772	000241			DEC	R3	; AS IS VALUE
3184	013774	006011			BNE	3#	
3185	013776	005303			TST	(R1)+	;BUMP PARAM POINTER
3186	014000	001374			MOV	NEWCYL,(R1)	;INSERT SB VALUE
3187	014002	005721			JSR	PC,RPTRES	;REPORT RESULTS
3188	014004	016711	167072		JSR	PC,RPTREM	;REPORT REMAINDER
3189	014010	004767	010364		MOV	(SP)+,R3	;RESTORE R3
3190	014014	004767	010566		MOV	(SP)+,R1	;RESTORE R1
3191	014020	012603			JSR	PC,CKERLM	;GO CHECK IF ERROR COUNT EXCEEDED
3192	014022	012601		ENDMSG			
3193	014024	004767	002414	L10007:			
3194	014030				TRAP	C#MSG	
	014030						
	014030	104423		BGNMSG	ERR9		
3195					INC	@ERRPOINT	;BUMP ERROR COUNT
3196	014032				MOV	R1,-(SP)	;STORE R1
3197	014032	005277	167120		JSR	PC,RPTOP	;REPORT OPERATION
3198	014036	010146			MOV	#3,(R1)+	;SET PARAM NUMBER
3199	014040	004767	007546		MOV	R3,(R1)+	;INSERT NAME ADD POINTER
3200	014044	012721	000003		MOV		
3201	014050	010321					

ERROR MESSAGES

```

3202 014052 010421          MOV      R4,(R1)+      ;SET IS VALUE
3203 014054 010521          MOV      R5,(R1)+      ;SET SB VALUE
3204 014056 004767 010316    JSR      PC,RPTRES      ;REPORT RESULTS
3205 014062 004767 010520    JSR      PC,RPTREM      ;REPORT REMAINDER
3206 014066 012601          MOV      (SP)+,R1       ;RESTORE R1
3207 014070 004767 002350    JSR      PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
3208 014074          ENDMSG
      014074          L10010:
      014074 104423          BGNMSG
3209 014076          TRAP     C#MSG
3210 014076 010146          ERR10
      MOV      R1,-(SP)      ;STORE R1
3211 014100 005767 166710    TST     MORECE           ;TEST IF 2ND BAD LINE
3212 014104 001051          BNE     3#              ;YES - SKIP
3213 014106 005277 167044    INC     @ERRPOINT        ;BUMP ERROR COUNT
3214 014112 004767 007474    JSR     PC,RPTOP         ;REPORT OPERATION
3215 014116          PRINTB  @FMT5,@BASADD,RLBAS,@DRVNAM,<B,RLDRV+1> ;REPORT ID
      014116 005046          CLR     -(SP)
      014120 156716 166707    BISB   RLDRV+1,(SP)
      014124 012746 006053    MOV     @DRVNAM,-(SP)
      014130 016746 166672    MOV     RLBAS,-(SP)
      014134 012746 006042    MOV     @BASADD,-(SP)
      014140 012746 011567    MOV     @FMT5,-(SP)
      014144 012746 000005    MOV     #5,-(SP)
      014150 010600          MOV     SP,R0
      014152 104414          TRAP   C#PNTB
      014154 062706 000014    ADD     #14,SP
3216 014160          PRINTB  @FMT14,@MRSLT,@MWORD,R3,@RESE3,(R4),@RESE4,(R5)
      014160 011546          MOV     (R5),-(SP)
      014162 012746 011304    MOV     @RESE4,-(SP)
      014166 011446          MOV     (R4),-(SP)
      014170 012746 011300    MOV     @RESE3,-(SP)
      014174 010346          MOV     R3,-(SP)
      014176 012746 006172    MOV     @MWORD,-(SP)
      014202 012746 005423    MOV     @MRSLT,-(SP)
      014206 012746 012040    MOV     @FMT14,-(SP)
      014212 012746 000010    MOV     #10,-(SP)
      014216 010600          MOV     SP,R0
      014220 104414          TRAP   C#PNTB
      014222 062706 000022    ADD     #22,SP
3217 014226 000421          BR      4#
3218 014230          3#: PRINTB  @FMT15,@MWORD,R3,@RESE3,(R4),@RESE4,(R5) ;REPORT DATA
      014230 011546          MOV     (R5),-(SP)
      014232 012746 011304    MOV     @RESE4,-(SP)
      014236 011446          MOV     (R4),-(SP)
      014240 012746 011300    MOV     @RESE3,-(SP)
      014244 010346          MOV     R3,-(SP)
      014246 012746 006172    MOV     @MWORD,-(SP)
      014252 012746 012072    MOV     @FMT15,-(SP)
      014256 012746 000007    MOV     #7,-(SP)
      014262 010600          MOV     SP,R0
      014264 104414          TRAP   C#PNTB
      014266 062706 000020    ADD     #20,SP
3219 014272 005267 166516    4#: INC     MORECE           ;INC COMPARE ERROR COUNT
3220 014276 012601          MOV     (SP)+,R1       ;RESTORE R1
3221 014300 004767 002140    JSR     PC,CKERLM      ;GO CHECK IF ERROR COUNT EXCEEDED
3222 014304          ENDMSG
      014304          L10011:

```


ERROR MESSAGES

3223	014304	104423	TRAP	C#MSG	
3224	014306		ENDMOD	.EVEN	
3225					
3226	014306		BGNMOD	HPTCODE	
3227	014306		BGNHW		
	014306	000006		.WORD	L10012-L#HW/2
3228	014310	174400		.WORD	174400
3229	014312	000160		.WORD	160
3230	014314	000240		.WORD	240
3231	014316	000001		.WORD	1
3232	014320	000000		.WORD	0
3233	014322	000001		.WORD	1
3234	014324		ENDHW		
	014324		L10012:		
3235	014324		ENDMOD		
3236					
3237	014324		BGNMOD	SPTCODE	
3238	014324		BGNSW		
	014324	000006		.WORD	L10013-L#SW/2
3239	014326	000000	MISWIW:	.WORD	0
3240					
3241					
3242					
3243					
3244					
3245					
3246					
3247	014330	000000	LOLIMW:	.WORD	0
3248	014332	000377	HILIMW:	.WORD	255.
3249	014334	000000	HEADW:	.WORD	0
3250	014336	000024	ERLIMW:	.WORD	20.
3251	014340	000012	DCLIMW:	.WORD	10.
3252	014342		ENDSW		
	014342		L10013:		
3253	014342		ENDMOD		
3254					
3255	014342		BGNMOD	DSPCODE	
3257	014342		DISPATCH		16
	014342	000020		.WORD	16
	014344	025072		.WORD	T1
	014346	025352		.WORD	T2
	014350	025560		.WORD	T3
	014352	027300		.WORD	T4
	014354	030202		.WORD	T5
	014356	030606		.WORD	T6
	014360	031574		.WORD	T7
	014362	032372		.WORD	T8
	014364	032460		.WORD	T9
	014366	033016		.WORD	T10
	014370	033376		.WORD	T11
	014372	034154		.WORD	T12
	014374	034530		.WORD	T13
	014376	034750		.WORD	T14
	014400	035230		.WORD	T15
	014402	035644		.WORD	T16
3262	014404		ENDMOD		

;CSR BASE ADDRESS DEFAULT
;VECTOR DEFAULT
;PRIORITY DEFAULT
;TYPE OF DRIVE, RL01=1, RL02=2
;DRIVE NUMBER DEFAULT
;RL11 CONTROLLER

;BIT 0 = USE ALL CYLINDERS
;BIT 1 = USE ALL SECTORS
;BIT 2 = EXECUTE DRIVE SELECT TEST
;BIT 3 = EXECUTE HEAD ALIGNMENT
;BIT 12 = HEAD SELECT SUPPLIED FLAG
;BIT 13 = HILIMIT SPECIFIED FLAG
;BIT 14 = LO LIMIT SPECIFIED FLAG
;BIT 15 = DO MANUAL INTERVENTION

;ERROR LIMIT
;COMPARE ERROR LIMIT

ERROR MESSAGES

```

3263
3264
3265 014404
3266 014404 000000
3267 014406 177777
3268 014410 000010
3269 014412
3270
3271
3272
3273 014412
3274 014412
3275 014412
    014412 012746 000340
    014416 012746 170000
    014422 012746 000140
    014426 012746 000003
    014432 104437
    014434 062706 000010
3276
3277 014440 005067 166500
3278
3279
3280
3281
3282 014444 000503
3283
3284 014446 012700 000120
3285 014452 104462
3286 014454 010067 166466
3287 014460 103004
3288 014462 012767 000001 166454
3289 014470 000451
3290
3291 014472 012737 014606 000004
3292 014500 005737 177546
3293
3294 014504 012767 000011 166432
3295
3296 014512 012737 014544 000100
3297
3298 014520 010146
3299 014522 010246
3300
3301 014524 005002
3302 014526 012737 000100 177546
3303
3304 014534 062702 000001
3305 014540 000240
3306 014542 000774
3307
3308 014544 012716 014552
3309 014550 000002
3310 014552 005037 177546
3311
3312 014556 012701 000246
3313
;LOAD PROTECTION TABLE
BGNPROT
.WORD 0 ;P-TABLE OFFSET OF CSR
.WORD -1 ;NOT A MASS-BUSS DRIVE
.WORD 10 ;P-TABLE OFFSET OF DRIVE
ENDPROT

.SBTTL INITIALIZATION CODE

BGNMOD INITCODE
BGNINIT
SETVEC #140,#170000,#340 ;ODT STARTING ADDR ;JSD REV A
MOV #340,-(SP)
MOV #170000,-(SP)
MOV #140,-(SP)
MOV #3,-(SP)
TRAP C#SVEC
ADD #10,SP

;CHECK FOR PRESENCE OF A CLOCK
PCLK: CLR CLKFLG ;CLEAR CLOCK FLAG ;JSD REV A
; ;JSD REV A
; REMOVE ALL REFERENCES TO AN EXTERNAL CLOCK. DELAY TIMING ;JSD REV A
; WILL BE DONE THROUGH INSTRUCTION LOOPS. ;JSD REV A
; ;JSD REV A
; BR NOCLK ;JUMP AROUND CLOCK INIT CODE ;JSD REV A
; ;JSD REV A
MOV #P,R0
TRAP C#CLCK
MOV R0,CLKADR
BCC NOPCLK
MOV #1,CLKFLG ;INDICATE PRESENCE OF A P-CLOCK
BR TCLK ;P CLOCK EXISTS, DO NOT USE L CLOCK.

NOPCLK: MOV #TSTCLK,#04 ;TEST FOR L CLOCK. IF NO CLOCK - SKIP.
TST #0177546

MOV #11,CLKFLG ;INDICATE THE PRESENCE OF AN L CLOCK.

MOV #LCLK,#0100 ;L CLOCK VECTOR POINTS TO LCLK.

MOV R1,-(SP) ;SAVE R1 AND R2 ON THE STACK.
MOV R2,-(SP)

CLR R2
MOV #100,#0177546 ;START THE L CLOCK.

10: ADD #1,R2 ;BUILD SOFTWARE LOOP. USE ADD TO SET FLAGS.
NOP
BR 10

LCLK: MOV #LCLK1,SP ;MODIFY THE STACK TO RETURN TO LCLK1.
RTI
LCLK1: CLR #0177546 ;STOP THE L CLOCK.

MOV #166.,R1 ;THIS IS THE DIVISOR TO GET 100 US.

```

INITIALIZATION CODE

```

3314 014562 005067 166362          CLR      LBASE
3315 014566 005267 166356          1$: INC      LBASE          ;LBASE IS THE APPROXIMATE NUMBER OF ITERATIONS
3316 014572 160102                    SUB      R1,R2          ;NEEDED TO GIVE 100 US.
3317 014574 100401                    BMI      2$
3318 014576 000773                    BR       1$
3319
3320 014600 012602          2$: MOV      (SP),R2          ;RESTORE R1 AND R2.
3321 014602 012601          MOV      (SP),R1
3322 014604 000403          BR       TCLK          ;SKIP RTI HANDLER
3323
3324 014606 012716 014614          TSTCLK: MOV     @TCLK,(SP)          ;ADJUST STACK FOR RTI
3325 014612 000002          RTI
3326 014614 005767 166324          TCLK:  TST     CLKFLG          ;IF THERE IS NO P OR L CLOCK, DO NOT DO THE
3327 014620 001015                    BNE      1$          ;TEST. PRINT A MESSAGE SAYING WHY THE TEST IS
3328 014622 012746 006664          MOV     @NOTST,-(SP)          ;ABORTED.
3329 014626 012746 011753          MOV     @FMT9,-(SP)
3330 014632 012746 000002          MOV     @2,-(SP)
3331 014636 010600          MOV     SP,RO
3332 014640 104417          TRAP   C:PNTF
3333 014642 062706 000006          ADD     @6,SP
3334 014646 012701 000200          MOV     @200,R1
3335 014652 000111          JMP     @R1
3336
3337          ;1$: SETPRI @340          ;SET PRI TO 7 TO INHIBIT INT'S          ;JSD REV A
3338 014654          1$:
3339 014654          NOCLK:          ;NEW LABEL          ;JSD REV A
3340 014654          SETPRI @PRI06          ;SET PRI TO 6 TO INHIBIT INT'S          ;JSD REV A
3341 014662 012700 000300          MOV     @PRI06,RO
3342 014664 104441          TRAP   C:SPRI
3343 014666 042767 100014 177432          MANUAL          ;CHECK IF MANUAL INTERVENTION ALLOWED
3344 014664 103403          TRAP   C:MANI
3345 014674 005067 166102          BCOMPLETE 2$          ;YES - SKIP
3346 014700 012700 000034          BCS     2$
3347 014706 103005          BIC     @MITEST!DRSELT!MDALIGN,MISWIW ;CLEAR ALL MANUAL
3348 014710 016767 165076 166452          ; INTERVENTION FLAGS
3349 014716 000167 000406          2$: CLR      SSINDX          ;CLEAR SUBROUTINE STACK INDEX
3350          ;"START" COMMAND SEQUENCE
3351 014722 012700 000040          4$: REDEF   @EF.START          ;CHECK IF START
3352 014730 103034          MOV     @EF.START,RO
3353          TRAP   C:REFG
3354          BCOMPLETE RESTART ;NO - SKIP
3355 014732 016767 165054 166134          BCC     RESTART
3356 014740 005067 166414          ; ON START INITIALIZE TO START AT FIRST DRIVE, CLEAR INTERNAL
3357 014744 012700 003160          ; PASS COUNT, AND ERROR COUNT.
3358 014750 012701 000100          MOV     L:UNIT,DRVCNT          ;SET UP UNIT COUNT
3359 014754 005020          RSTRT: CLR     PASNUM          ;CLEAR PASS NUMBER
3360 014756 005301          MOV     @ERRCNT,RO
3361          MOV     @64,,R1          ;GET A COUNT
3362          1$: CLR     (RO),          ;CLEAR ERROR COUNTER STORAGE AREA
3363          DEC     R1
    
```

INITIALIZATION CODE

```

3361 014760 001375          BNE      1$          ;LOOP TILL ALL CLEARED
3362 014762 012767 003156 166166    MOV      #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
3363 014770 012767 177777 166364    MOV      @-1,PSETNM      ;SET PARAM SELECT TO INITIAL VALUE
3364 014776 012767 177777 166004    MOV      @-1,HADONE      ;PRESET HEAD ALIGN DONE FLAG
3365 015004 032767 040000 177314    LAB:    BIT      @LOCYL,MISWIW ;TEST IF LO LIMIT SET
3366 015012 001002          BNE      5$          ;YES - SKIP
3367 015014 005067 177310    CLR      LOLIMW         ;ELSE CLEAR LO LIMIT
3368 015020 000432          BR       SETDON
3369 015022          ;
3370 015022          READEF   #EF.RESTART     ;CHECK IF RESTART
          015022 012700 000037    MOV      #EF.RESTART,RO
          015026 104447    TRAP    C$REFG
3371 015030          BCOMPLETE RSTRT      ;NO - SKIP
          015030 103743    BCS     RSTRT
          ;"CONTINUE" COMMAND SEQUENCE
3372          CONTINUE:
3373 015032          READEF   #EF.CONTINUE  ;TEST IF CONTINUE
3374 015032          MOV      #EF.CONTINUE,RO
          015032 012700 000036    TRAP    C$REFG
          015036 104447    BCOMPLETE PWCON
3375 015040          BCS     PWCON
          015040 103533    ; ON CONTINUE PICK UP UNIT LAST UNDER TEST
3376          ;
3377 015042          READEF   #EF.NEW       ;CHECK IF STARTING NEW PASS
          015042 012700 000035    MOV      #EF.NEW,RO
          015046 104447    TRAP    C$REFG
3378 015050          BCOMPLETE PASNEW
          015050 103403    BCS     PASNEW
3379 015052          NXTPAS:
3380 015052          TST      DRVCNT      ;TEST IF ALL UNITS CHECKED
          005767 166016    BNE      SETDON      ;NO - SKIP
3381 015056          PASNEW: INC      PASNUM    ;ELSE BUMP PASS COUNT
          001013          MOV      #ERRCNT-2,ERRPOINT ;INIT ERROR POINTER
3382 015060          MOV      L$UNIT,DRVCNT ;GET ALL DRIVES
          005267 166274    MOV      @-1,PSETNM    ;SET PARAM SELECT TO INITIAL
3383 015064          SETDON: INC     PSETNM   ;NEXT SET OF PARAMETERS
          012767 164714 165774    DEC     DRVCNT        ;DOWN COUNT DRIVE TOTAL
          012767 177777 166254    ADD     @2,ERRPOINT   ;UPDATE THE ERROR POINTER
3384 015072          MOV      PSETNM,RO    ;SET UP TO GET PARAMETERS
          016767 164714 165774    MOV     @RLBAS,R2    ;GET POINTER TO RL11 BASE ADDRESS
3385 015100          GPHARD  RO,R1
          012767 177777 166254    TRAP   C$GPHRD
3386 015106          MOV      RO,R1
          005267 166250    BCOMPLETE 7$          ;SKIP IF GOOD PARAM
          015112 005367 165756    BCS     7$
3387 015112          TST      PWRFLG      ;RECENT POWER FAILURE
          005367 165756    BEQ     NXTPAS      ;NO
3388 015116          DEC     PWRFLG      ;ACCOUNT FOR DRIVE
          062767 000002 166032    BR      NXTPAS
3389 015124          ;MOVE P-TABLE CONTENTS TO LOCAL STORAGE
          016700 166232    7$:  MOV     (R1), (R2)    ;STORE CSR
3390 015130          MOV     (R1), (R2)    ;STORE VECTOR
          012702 003026    TST     (R1)         ;BUMP PAST PRIORITY
3391 015134          MOV     (R1), T.DRIVE ;STORE DRIVE TYPE
          015134 104442    MOV     (R1), (R2)
          015136 010001    CMP     @1, T.DRIVE
3392 015140          BEQ     65$
          015140 103406    ;INITIALIZE RL02 PARAMETERS
3393 015142          65$:
          005767 166222
3394 015146          001741
3395 015150          005367 166214
3396 015154          000736
3397
3398 015156          012122
3399 015160          012122
3400 015162          005721
3401 015164          012167 165106
3402 015170          012122
3403 015172          022767 000001 165076
3404 015200          001426
3405

```

INITIALIZATION CODE

```

3406 015202 012767 000776 165076      MOV      #510.,NXTHL
3407 015210 012767 000777 165064      MOV      #511.,HLMTW
3408 015216 012767 001000 165064      MOV      #512.,GBND
3409 015224 012767 177600 165060      MOV      #177600,CAMSK
3410 015232 012767 177600 165054      MOV      #177600,DIRMSK
3411 015240 012767 177600 165050      MOV      #177600,HDCYL
3412 015246 012767 177000 165030      MOV      #177000,CLRBYT
3413 015254 000425
3414
3415 015256 012767 000377 165016      ;INITIALIZE RLO1 PARAMETERS
65:  MOV      #255.,HLMTW
3416 015264 012767 000400 165016      MOV      #256.,GBND
3417 015272 012767 077600 165012      MOV      #77600,CAMSK
3418 015300 012767 077600 165006      MOV      #77600,DIRMSK
3419 015306 012767 077600 165002      MOV      #77600,HDCYL
3420 015314 012767 000376 164764      MOV      #254.,NXTHL
3421 015322 012767 177400 164754      MOV      #177400,CLRBYT
3422
3423 015330 032767 020000 176770 PWCON:  BIT      #HICYL,MISWIW
3424 015336 001003      BNE      1#
3425 015340 016767 164736 176764      MOV      HLMTW,HILIMW
3426 015346      1#:  SETVEC  RLVEC,#INTHLR,#340      ;SET UP INTERRUPT VECTOR FOR DRIVE
      MOV      #340,-(SP)
      MOV      #INTHLR,-(SP)
      MOV      RLVEC,-(SP)
      MOV      #3,-(SP)
      TRAP    C#SVEC
      ADD     #10,SP
3427 015374      SETPRI  #0      ;SET PRIORITY TO 0 TO ALLOW INTERRUPTS
      MOV      #0,R0
      TRAP    C#SPRI
3428 015402 016702 165420      MOV      RLBAS,R2      ;SET RL11 BASE ADDRESS POINTER
3429
3431
3432 015406      MANUAL
      TRAP    C#MANI      ;MANUAL INTERVENTION ALLOWED?
3433 015410 104450      BNCOMPLETE 4#      ;NO
      BCC     4#
3434
3435 015412 005767 165742      TST     PASNUM      ;YES, CHECK PASS NUMBER
3436 015416 001001      BNE     4#          ;NOT FIRST PASS, NEED DRIVE UP
3437 015420 000514      BR      8#          ;FIRST PASS, PROGRAM WILL INSTRUCT USER
3438
3440      ;CHECK IF POWER FAILURE WAIT IS NEEDED
3441
3442 015422 005767 165742      4#:  TST     PWRFLG      ;NEEDED?
3443 015426 001511      BEQ     8#          ;NO, SKIP
3444
3445 015430 016705 165376      MOV      RLDV,R5      ;DRIVE SELECT
3446 015434 052705 000200      BIS      #CRDYMSK,R5  ;SET CRDY
3447 015440 010562 000000      MOV      R5,RLCS(R2)  ;SELECT DRIVE
3448 015444 012701 000170      MOV      #120.,R1     ;INITIALIZE WAIT COUNT
3449 015450 032762 000001 000000 9#:  BIT      #DRDYMSK,RLCS(R2) ;DRIVE UP YET
3450 015456 001075      BNE     8#          ;YES START TEST
3451
3452 015460      WAITMS  10.        ;WAIT A SECOND
3453 015524 005301      DEC     R1          ;SIXTY GONE BY
3454 015526 001350      BNE     9#          ;NO
    
```

INITIALIZATION CODE

```

3455 015530          PRINTF  #FMT24,#NOPWR ;REPORT "DRV DID NOT REC'R FROM PWR FAIL"
      015530 012746 006060  MOV    #NOPWR,-(SP)
      015534 012746 012373  MOV    #FMT24,-(SP)
      015540 012746 000002  MOV    #2,-(SP)
      015544 010600          MOV    SP,RO
      015546 104417          TRAP   C:PNTF
      015550 062706 000006  ADD    #6,SP
3456 015554          PRINTF  #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;REPORT DRIVE UNIBUS
      015554 005046          CLR    -(SP)
      015556 156716 165251  BISB  RLDRV+1,(SP)
      015562 012746 006053  MOV    #DRVNAM,-(SP)
      015566 016746 165234  MOV    RLBAS,-(SP)
      015572 012746 006042  MOV    #BASADD,-(SP)
      015576 012746 011567  MOV    #FMT5,-(SP)
      015602 012746 000005  MOV    #5,-(SP)
      015606 010600          MOV    SP,RO
      015610 104417          TRAP   C:PNTF
      015612 062706 000014  ADD    #14,SP
3457                                     ;/ADDRESS AND DRIVE NUMBER
3458 015616          PRINTF  #FMT3 ;NEW LINE
      015616 012746 011553  MOV    #FMT3,-(SP)
      015622 012746 000001  MOV    #1,-(SP)
      015626 010600          MOV    SP,RO
      015630 104417          TRAP   C:PNTF
      015632 062706 000004  ADD    #4,SP
3459 015636          DODU    PSETNM ;DO DROP UNIT ON DRIVE
      015636 016700 165520  MOV    PSETNM,RO
      015642 104451          TRAP   C:DODU
3460 015644          DOCLN  ;INVOKE CLEAN-UP CODE TO RESTORE DRIVE
      015644 104444          TRAP   C:DCLN
3461                                     ;/TO STATIC STATE
3462 015646 005067 165266  CLR    ERRVEC ;CLEAR ERROR VECTOR
3463
3464 015652          B#:
3465
3466 015652          ENDINIT
      015652          L10015:
      015652 104411          TRAP   C:INIT
3467
3468 015654          ENDMOD
3469
3470          .SBTTL  AUTO DROP SECTION
3471
3472          ;THE AUTO DROP SECTION IS INVOKED BY THE DIAGNOSTIC SUPERVISOR WHENEVER THE
3473          ;"ADR" FLAG IS SET BY THE OPERATOR. IT IS EXECUTED AFTER THE INITIALIZATION
3474          ;CODE AND CHECKS THE DRIVE TO DETERMINE IF IT IS READY TO RECEIVE A COMMAND.
3475          ;IF THE DRIVE IS NOT READY IT IS DROPPED FROM THE TEST CYCLE AND THE NEXT
3476          ;DRIVE IS ACCESSED. IF THE DRIVE IS READY THE HARDWARE TESTS ARE PERFORMED
3477          ;AFTER WHICH THE NEXT DRIVE IS ACCESSED.
3478
3479          BGNAUTO
3480 015654 005067 165506  CLR    TRPFLG ;CLEAR TRAP FLAG
3481 015660          SETVEC  ERRVEC,#TRPHAN,#340 ;SET UP TRAP VECTOR TO DETECT
      015660 012746 000340  MOV    #340,-(SP)
      015664 012746 016436  MOV    #TRPHAN,-(SP)
      015670 016746 165244  MOV    ERRVEC,-(SP)
      015674 012746 000003  MOV    #3,-(SP)
    
```

AUTO DROP SECTION

	015700	104437				
	015702	062706	000010			
3482				TRAP	C#SVEC	
3483				ADD	#10,SP	
						;/NON-EXISTENT CONTROLLER UNIBUS
						;/ADDRESS
3484	015706	016702	165114	MOV	RLBAS,R2	;/GET RL11 BASE ADDRESS
3485	015712	005762	000000	TST	RLCS(R2)	;/ACCESS DRIVE CONTROLLER UNIBUS ADDRESS
3486	015716	005767	165444	TST	TRPFLG	;/DID TRAP OCCUR?
3487	015722	001447		BEQ	1#	;/BRANCH TO CHECK DRIVE IF TRAP DID NOT OCCUR
3488	015724			PRINTF	#FMT24,#NOCTLR	;/ELSE, PRINT MSG. "DRV DROPPED - NO CNTLR"
	015724	012746	006754	MOV	#NOCTLR,-(SP)	
	015730	012746	012373	MOV	#FMT24,-(SP)	
	015734	012746	000002	MOV	#2,-(SP)	
	015740	010600		MOV	SP,RO	
	015742	104417		TRAP	C#PNTF	
	015744	062706	000006	ADD	#6,SP	
						;/PRINT DRIVE INFORMATION
3489				PRINTF	#FMT5,#BASADD,RLBAS,#DRVNUM,<B,RLDRV+1>	
3490	015750			CLR	-(SP)	
	015750	005046		BISB	RLDRV+1,(SP)	
	015752	156716	165055	MOV	#DRVNUM,-(SP)	
	015756	012746	006053	MOV	RLBAS,-(SP)	
	015762	016746	165040	MOV	#BASADD,-(SP)	
	015766	012746	006042	MOV	#FMT5,-(SP)	
	015772	012746	011567	MOV	#5,-(SP)	
	015776	012746	000005	MOV	SP,RO	
	016002	010600		TRAP	C#PNTF	
	016004	104417		ADD	#14,SP	
	016006	062706	000014	PRINTF	#FMT3	
3491	016012			MOV	#FMT3,-(SP)	
	016012	012746	011553	MOV	#1,-(SP)	
	016016	012746	000001	MOV	SP,RO	
	016022	010600		TRAP	C#PNTF	
	016024	104417		ADD	#4,SP	
	016026	062706	000004	DODU	PSETNM	;/DO DROP UNIT ON DRIVE
3492	016032			MOV	PSETNM,RO	
	016032	016700	165324	TRAP	C#DODU	
	016036	104451		BR	2#	;/BRANCH TO EXIT
3493	016040	000460		MOV	RLDRV,R5	;/ELSE, GET DRIVE NUMBER
3494	016042	016705	164764	BIS	#CRDYMSK,R5	;/SET CONTROLLER READY
3495	016046	052705	000200	MOV	R5,RLCS(R2)	;/LOAD IN THE DRIVE NUMBER
3496	016052	010562	000000	BIT	#DRDYMSK,RLCS(R2)	;/IS DRIVE READY?
3497	016056	032762	000001	BNE	2#	;/BRANCH TO PERFORM TESTS IF DRIVE IS READY
3498	016064	001046		PRINTF	#FMT24,#NOTRDY	;/PRINT MSG. "DRV DROPPED - NOT RDY"
3499	016066			MOV	#NOTRDY,-(SP)	
	016066	012746	007003	MOV	#FMT24,-(SP)	
	016072	012746	012373	MOV	#2,-(SP)	
	016076	012746	000002	MOV	SP,RO	
	016102	010600		TRAP	C#PNTF	
	016104	104417		ADD	#6,SP	
	016106	062706	000006			;/PRINT DRIVE INFORMATION
3500				PRINTF	#FMT5,#BASADD,RLBAS,#DRVNUM,<B,RLDRV+1>	
3501	016112			CLR	-(SP)	
	016112	005046		BISB	RLDRV+1,(SP)	
	016114	156716	164713	MOV	#DRVNUM,-(SP)	
	016120	012746	006053	MOV	RLBAS,-(SP)	
	016124	016746	164676	MOV	#BASADD,-(SP)	
	016130	012746	006042	MOV	#FMT5,-(SP)	
	016134	012746	011567	MOV		

1#:

000000

AUTO DROP SECTION

```

016140 012746 000005      MOV      #5,-(SP)
016144 010600      MOV      SP,RO
016146 104417      TRAP     C#PNTF
3502 016150 062706 000014      ADD      #14,SP
016154 012746 011553      PRINTF  #FMT3
016160 012746 000001      MOV      #FMT3,-(SP)
016164 010600      MOV      #1,-(SP)
016166 104417      MOV      SP,RO
016170 062706 000004      TRAP     C#PNTF
3503 016174 016700 165162      ADD      #4,SP
016174 016700 165162      DODU     PSETNM          ;DO DROP UNIT ON DRIVE
016200 104451      MOV      PSETNM,RO
3504 016202 016700 164732      TRAP     C#DODU
016202 016700 164732      CLRVEC  ERRVEC          ;RELEASE THE ERROR VECTOR
016206 104436      MOV      ERRVEC,RO
3505 016210 104461      TRAP     C#CVEC
016210 104461      TRAP     C#AUTO
3506 016210 104461      TRAP     C#AUTO
3507 016212      BGNMOD  CLNCODE
3508 016212      BGNCLN
3509
3510 016212      SETVEC  ERRVEC,#TRPHAN,#340
016212 012746 000340      MOV      #340,-(SP)
016216 012746 016436      MOV      #TRPHAN,-(SP)
016222 016746 164712      MOV      ERRVEC,-(SP)
016226 012746 000003      MOV      #3,-(SP)
016232 104437      TRAP     C#SVEC
016234 062706 000010      ADD      #10,SP
3511
3512
3513 016240      SETPRI  #7              ;SET PRIORITY TO 7
016240 012700 000300      SETPRI  #PRI06         ;SET PRIORITY TO 6
016244 104441      MOV      #PRI06,RO
3514 016246 032762 000200 000000 2#      TRAP     C#SPRI
016254 001407      BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
3515 016254 001407      BEQ     3#              ;NO LOOP UNTIL READY
3516 016256 056762 164550 000000      BIS      RLDRV,RLCS(R2) ;SET DRIVE NUMBER
3517 016264 032762 000001 000000      BIT      #DRDYMSK,RLCS(R2) ;TEST IF DRIVE BUSY
3518 016272 001022      BNE     5#              ;NO - SKIP
3519 016274      WAITMS 3              ;WAIT 300 MS
3520 016340      CLRVEC  RLVEC          ;RELEASE DRIVE VECTOR
016340 016700 164464      MOV      RLVEC,RO
016344 104436      TRAP     C#CVEC
3521 016346 005767 165016      TST     PWRFLG          ;PWR FAIL SET
3522 016352 001402      BEQ     7#              ;NO
3523 016354 005367 165010      DEC     PWRFLG
3524 016360      CLRVEC  ERRVEC
016360 016700 164554      MOV      ERRVEC,RO
016364 104436      TRAP     C#CVEC
3525 016366      ENDCLN
016366 104412      L10017: TRAP     C#CLEAN
3526
3527 016370      BGNDU
3528 016370 000240      NOP
3529 016372      ENDDU

```


AUTO DROP SECTION

```

016372          L10020:
016372 104453   TRAP    C#DU
3530
3531 016374     ENDMOD
3532
3533           .SBTTL  INTERRUPT SERVICE ROUTINES
3534
3535 016374     BGNSRV  INTHLR
3536           ;INTERRUPT HANDLER FOR DRIVE ABORTS WAIT TIMER AND STORES ALL RL11 REGISTERS
3537           ;
3538 016374 012267 164444   CLR    DLYCNT          ;CLEAR UNELAPSED DELAY COUNT          ;JSD REV A
3539 016400 012267 164442   MOV    (R2)+,T.CS      ;STORE RL REGISTERS
3540 016404 012267 164440   MOV    (R2)+,T.BA
3541 016410 011267 164436   MOV    (R2)+,T.DA
3542 016414 012767 177777   MOV    (R2),T.MP
3543 016422 016702 164400   MOV    #-1,DONE       ;SET DONE FLAG
3544 016426           MOV    RLBAS,R2          ;RESTORE R2
016426           ENDSRV
016426 000002     L10021:
3545           RTI
3546           ;INTERRUPT SERVICE ROUTINE FOR P-CLOCK DECREMENTS DELAY COUNTER AT 100-MICROSECOND
3547           ;TIME INTERVALS
3548 016430     BGNSRV  CLKINT
3549 016430 005367 164506   DEC    DLYCNT          ;DECREMENT CLOCK DELAY COUNTER
3550 016434     ENDSRV
016434           L10022:
016434 000002     RTI
3551
3552           ;INTERRUPT SERVICE ROUTINE SETS TRAP FLAG WHEN A NON-EXISTENT UNIBUS ADDRESS IS
3553           ;ACCESSED
3554 016436     BGNSRV  TRPHAN
3555 016436 005267 164724   INC    TRPFLG          ;INDICATE THAT TRAP OCCURRED
3556 016442     ENDSRV
016442           L10023:
016442 000002     RTI
3557
3558           .SBTTL  GLOBAL SUBROUTINES
3559
3560 016444     BGNMOD  GLBSUB
3561
3562           ;
3563           ; ERROR LIMIT CHECKING ROUTINE
3564 016444 027767 164506 175664 CKERLM: CMP    $ERRPOINT,ERLIMW ;TEST IF ERROR LIMIT EXCEEDED
3565 016452 002453           BLT    1#              ;NO - SKIP
3566 016454           INLOOP TRAP    C#INLP          ;CHECK IF IN ERROR LOOP
3567 016456           BCOMPLETE 1#              ;YES - SKIP
3568 016460           BCS    1#
3568 016460 012746 011225   PRINTF #FMT25,ERLIMW,#MEXERS ;PRINT MSG. "OVER ERROR LIMIT - UNIT DROPPED"
3568 016464 016746 175646   MOV    #MEXERS,-(SP)
3568 016470 012746 012400   MOV    ERLIMW,-(SP)
3568 016474 012746 000003   MOV    #FMT25,-(SP)
3568 016500 010600           MOV    #3,-(SP)
3568 016502 104417           MOV    SP,R0
3568 016504 062706 000010   TRAP  C#PNTF
3569 016510           ADD    #10,SP
3569 016510           PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1> ;PRINT DRIVE INFORMATION

```

GLOBAL SUBROUTINES

```

016510 005046          CLR      -(SP)
016512 156716 164315  BISB    RLDRV+1,(SP)
016516 012746 006053  MOV     @DRVNAM,-(SP)
016522 016746 164300  MOV     RLBAS,-(SP)
016526 012746 006042  MOV     @BASADD,-(SP)
016532 012746 011567  MOV     @FMT5,-(SP)
016536 012746 000005  MOV     #5,-(SP)
016542 010600          MOV     SP,RO
016544 104417          TRAP   C:PNTF
016546 062706 000014  ADD     #14,SP
3570 016552          PRINTF @FMT3
016552 012746 011553  MOV     @FMT3,-(SP)
016556 012746 000001  MOV     #1,-(SP)
016562 010600          MOV     SP,RO
016564 104417          TRAP   C:PNTF
016566 062706 000004  ADD     #4,SP
3571 016572          DODU   PSETNM          ;DROP DRIVE
016572 016700 164564  MOV     PSETNM,RO
016576 104451          TRAP   C:DODU
3572 016600          DOCLN          ;GO TO CLEAN UP
016600 104444          TRAP   C:DCLN
3573 016602 000207          1$:   RTS     PC
3574
3575          ; READ AND STORE ALL RL11 REGISTERS
3576 016604 016267 000000 164232 ; READRL: MOV    RLCSR(R2),T.CS ;GET CS REG
3577 016612 016267 000002 164226 ;        MOV    RLBA(R2),T.BA ;GET BUS ADDRESS REG
3578 016620 016267 000004 164222 ;        MOV    RLDA(R2),T.DA ;GET DISK ADDRESS
3579 016626 016267 000006 164216 ;        MOV    RLMP(R2),T.MP ;GET MULTI-PURPOSE REG
3580 016634 000207          RTS     PC ;RETURN
3581
3582          ; WAIT FOR CONTROLLER TIMEOUT TO FORCE INTERRUPT ROUTINE
3583 016636 011646          ; WAITIN: MOV    (SP),-(SP) ;MAKE ROOM FOR ERROR POINTER
3584 016640 005066 000002          CLR    2(SP) ;CLEAR FOR POINTER
3585 016644 032762 000200 000000          BIT    @CRDYMSK,RLCSR(R2) ;TEST IF CONTROLLER READY
3586 016652 001420          BEQ   4$ ;NO - SKIP TO WAIT
3587 016654 004767 177724          JSR   PC,READRL ;READ ALL RL REGS
3588 016660 005767 164122          TST   DONE ;TEST IF INTERRUPT OCCURRED
3589 016664 001444          BEQ   5$ ;NO - GO SET NO INTERRUPT ERR FLAG
3590 016666 012766 006200 000002 1$:   MOV    @MTOSLOW,2(SP) ;ELSE SET TOO SLOW ERROR POINTER
3591 016674 032767 002000 164142          BIT    @OPIERR,T.CS ;TEST IF OPI SET
3592 016702 001403          BEQ   2$ ;NO - SKIP
3593 016704 012766 006220 000002          MOV    @MDRRES,2(SP) ;SET MESSAGE FOR NO DRIVE RESPONSE
3594 016712 000207          2$:   RTS     PC ;RETURN
3595          ;4$:   MOV    #1,DLYCNT ;INITIALIZE DELAY COUNT ;JSD REV A
3596          ;      ASL    DLYCNT ;MULTIPLY BY 2 ;JSD REV A
3597          ;      ASL    DLYCNT ;MULTIPLY BY 2 AGAIN ;JSD REV A
3598          ;      MOV    #10,(PC)+ ;IMPLEMENT TIME DELAY LOOP ;JSD REV A
3599          ;      .WORD 0 ;JSD REV A
3600          ;      MOV    L#DLY,(PC)+ ;JSD REV A
3601          ;      .WORD 0 ;JSD REV A
3602          ;      DEC    -6(PC) ;JSD REV A
3603          ;      BNE    -.4 ;JSD REV A
3604          ;      DEC    -22(PC) ;JSD REV A
3605          ;      BNE    .-20 ;JSD REV A
3606 016714          4$:   WAITUS 600. ;WAIT 6 MS ;JSD REV A
3607 016744 032762 000200 000000          BIT    @CRDYMSK,RLCS(R2) ;TEST IF READY NOW SET
3608 016752 001006          BNE    3$ ;YES - SKIP

```

GLOBAL SUBROUTINES

```

3609 016754 004767 177624      JSR    PC,READRL      ;READ RL REGS
3610 016760 012766 006273 000002  MOV    @MCONHNG,2(SP) ;SET MESSAGE FOR CONTROLLER HUNG
3611 016766 000751          BR     2#             ;SKIP
3612 016770 005767 164012      3# :   TST    DONE      ;ELSE CHECK IF INTERRUPT OCCURRED
3613 016774 001334          BNE    1#             ;YES - SKIP TO SET TOO SLOW
3614 016776 004767 177602      5# :   JSR    PC,READRL      ;READ RL REGS
3615 017002 012766 006240 000002  MOV    @MNOINT,2(SP) ;ELSE SET NO INTERRUPT FLAG
3616 017010 000740          BR     2#             ;GO TO RETURN
3617
3618
3619 017012 005067 163766      ;TSTINT: CLR    OPFLAG      ;CLEAR OPERATION FLAGS
3620 017016 105067 164343      CLRB   NOERCT        ;RESET INHIBIT ERROR COUNTING
3621 017022 005067 163766      CLR    MORECE        ;RESET MORE COMPARE ERRORS
3622 017026 000207          RTS     PC
3623
3624
3625 017030 016746 164072      ;GSTATR: MOV    TEMP4,-(SP) ;STORE TEMP4
3626 017034 012767 000013 164064  MOV    @GETSTAT!DRSET,TEMP4 ;SET FOR RESET
3627 017042 000412          BR     GSTATG
3628 017044 016746 164056      GSTATC: MOV    TEMP4,-(SP) ;STORE TEMP4
3629 017050 012767 000003 164050  MOV    @GETSTAT,TEMP4 ;SET FOR NO RESET
3630 017056 000404          BR     GSTATG
3631 017060 016746 164042      GSTAT:  MOV    TEMP4,-(SP) ;STORE TEMP4
3632 017064 005067 164036      CLR    TEMP4         ;SET FOR SAVE L. AND T. REGS
3633 017070 010346      GSTATG: MOV    R3,-(SP) ;STORE R3
3634 017072 016703 163704      MOV    SSINDX,R3    ;GET SUBROUTINE INDEX
3635 017076 005723          TST    (R3)+        ;BUMP IT FOR NEXT ENTRY
3636 017100 016663 000004 002404  MOV    4(SP),SUBSTK(R3) ;INSERT THIS CALL
3637 017106 162763 000004 002404  SUB    @4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
3638 017114 010367 163662          MOV    R3,SSINDX   ;STORE IT BACK
3639 017120 010046          MOV    R0,-(SP)    ;STORE R0
3640 017122 010146          MOV    R1,-(SP)    ;STORE R1
3641 017124 012767 000002 163664  MOV    @2,ERRSWI    ;SET FOR NO ERROR RETURN
3642 017132 032767 000010 163766  BIT    @DRSET,TEMP4 ;TEST IF DRIVE RESET
3643 017140 001513          BEQ    11#         ;NO - SKIP
3644 017142 032762 040000 000000  BIT    @DRVERR,RLCS(R2) ;TEST IF DRIVE ERROR SET
3645 017150 001422          BEQ    49#         ;NO - SKIP
3646 017152          WAITMS 3         ;WAIT FOR DRIVE TO SETTLE
3647
3648 017216 012701 000062      .NLIST ME
3649 017222 004767 177632      49# :   MOV    @50,R1     ;INITIALIZE WAIT COUNTER
3650 017226 020034          50# :   JSR    PC,GSTAT   ;GET DRIVE STATUS
3651 017230 032767 000001 163606  BIT    @DRDYMSK,T.CS ;TEST IF DRIVE READY
3652 017236 001072          BNE    5#         ;YES - GO DO CLEAR
3653 017240 032767 000020 163604  BIT    @HOSTAT,T.MP ;ELSE TEST IF HEADS OUT
3654 017246 001010          BNE    51#         ;YES - BYPASS RELOAD WAIT FLAG SETTING
3655 017250 032767 144000 163574  BIT    @SPDSTAT!HCESTAT!WDESTAT,T.MP ;TEST IF DRIVE HAS ERROR
3656
3657
3658 017256 001462          BEQ    5#         ;NO - SKIP
3659 017260 052767 040000 163516  BIS    @RELDWT,OPFLAG ;ELSE SET WAIT FLAG
3660 017266 000456          BR     5#         ;SKIP TO CLEAR
3661 017270 032767 040000 163546  51# :   BIT    @DRVERR,T.CS ;TEST IF DRIVE ERROR NOW
3662 017276 001052          BNE    5#         ;YES - SKIP TO CLEAR
3663 017300          WAITMS 1         ;WAIT FOR DRIVE TO GET ERROR, READY, OR HEADS OUT
3664 017344 005301          DEC    R1         ;DEC WAIT COUNTER
3665 017346 001325          BNE    50#         ;IF NOT DONE, LOOP

```

GLOBAL SUBROUTINES

```

3666 017350 012703 011103      MOV      #MUNDEF,R3      ;MESSAGE FOR UNDEFINED STATE
3667 017354 104456      ERRHRD  10001.,,ERR1
      017356 023421      TRAP    C$ERRHRD
      017360 000000      .WORD  10001
      017362 012464      .WORD  0
3668 017364 000167 000440      .WORD  ERR1
3669 017370 005767 163532      JMP     14#             ;EXIT
      11#: TST     TEMP4          ;TEST IF SAVE REGISTERS
3670 017374 001013      BNE     5#             ;NO SKIP
3671 017376 012701 000004      MOV     #4,R1          ;SET SAVE COUNT
3672 017402 012703 003044      MOV     @L.MP+2,R3     ;SET ADDRESS OF FIRST SAVE
3673 017406 014346      8#:    MOV     -(R3),-(SP)  ;PUT REG ON STACK
3674 017410 005301      DEC     R1             ;DEC COUNT
3675 017412 001375      BNE     8#             ;LOOP UNTIL ALL SAVED
3676 017414 012767 000003 163416      MOV     #GETSTAT,L.DA  ;SET FOR GET STATUS
3677 017422 000403      BR      6#             ;SKIP
3678 017424 016767 163476 163406      5#:    MOV     TEMP4,L.DA   ;INSERT PRESET FOR STATUS
      6#:
3680 017432 005067 163350      CLR     DONE           ;CLEAR INTERRUPT FLAG
3681 017436 016767 163370 163370      MOV     RLDRV,L.CS     ;SET UP TO GET STATUS
3682 017444 042767 002000 163362      BIC     #BIT10,L.CS    ;CLEAR FOR DRIVE 4 - 7 SPEC'D
3683 017452 052767 000104 163354      BIS     #GTSTAT,L.CS
3684 017460 016762 163354 000004      MOV     L.DA,RLDA(R2)  ;LOAD RL REGS
3685 017466 016762 163342 000000      MOV     L.CS,RLCSR(R2) ;LOAD CS REG
3686 017474      WAITUS 1              ;WAIT FOR INTERRUPT
3687 017524 005767 163256      TST     DONE           ;CHECK IF INTERRUPT OCCURRED
3688 017530 001530      BEQ     1#             ;NO - SKIP
3689 017532 016767 163314 163320      4#:    MOV     T.MP,T.STAT   ;STORE MP REGISTER
3690 017540 042767 177770 163312      BIC     #C<STAMSK>,T.STAT ;CLEAR ALL BUT STATE
3691 017546 032767 000010 163264      BIT     #DRSET,L.DA    ;TEST IF RESET WAS SPECIFIED
3692 017554 001527      BEQ     3#             ;NO - SKIP TO EXIT
3693 017556 032767 040000 163220      BIT     #RELDWT,OPFLAG ;TEST IF RELOAD WAIT FLAG SET
3694 017564 001444      BEQ     12#            ;NO - SKIP
3695 017566 012701 001130      MOV     #600.,R1       ;INITIALIZE WAIT COUNTER
3696 017572 032762 000001 000000      13#:   BIT     #DRDYMSK,RLCS(R2) ;TEST IF DRIVE NOW READY
3697 017600 001036      BNE     12#            ;YES - SKIP
3698 017602      WAITMS 1              ;CALL WAIT
3699 017646 005301      DEC     R1             ;DEC COUNT
3700 017650 001350      BNE     13#            ;LOOP IF NOT 0
3701 017652 004767 177202      JSR     PC,GSTAT       ;GET DRIVE STATUS
3702 017656 020034      3#
3703 017660 012703 011150      MOV     #MRLFAL,R3     ;SET RESULT MESSAGE POINTER
3704 017664      ERRHRD 10003.,,ERR1
      017666 023423      TRAP    C$ERRHRD
      017670 000000      .WORD  10003
      017672 012464      .WORD  0
3705 017674 000455      .WORD  ERR1
3706 017676      12#:   BR      14#            ;GO TO EXIT
3707 017726 004767 177126      WAITUS 10             ;WAIT
      JSR     PC,GSTAT     ;GET DRIVE STATUS
3708 017732 020034      3#
3709 017734 032767 100000 163102      BIT     #ANYERR,T.CS   ;TEST IF ANY ERROR
3710 017742 001434      BEQ     3#             ;NO - SKIP
3711 017744 032767 001000 163100      BIT     #VCSTAT,T.MP   ;CHECK IF VOLUME CHECK RESET
3712 017752 001403      BEQ     7#             ;YES SKIP
3713 017754 012703 006327      MOV     #VCNRST,R3     ;SET REASON POINTER
3714 017760 000417      BR      2#             ;EXIT

```

GLOBAL SUBROUTINES

```

3715 017762 032767 040000 163054 7#: BIT #DRVERR,T.CS ;CHECK IF DRIVE ERROR
3716 017770 001405 BEQ 9# ;NO - SKIP
3717 017772 ERRHRD 10004...ERR6
      017772 104456 TRAP C#ERHRD
      017774 023424 .WORD 10004
      017776 000000 .WORD 0
      020000 012766 .WORD ERR6
3718 020002 000412 BR 14# ;EXIT
3719 020004 012703 006350 9#: MOV #UNXERR,R3 ;SET REASON POINTER
3720 020010 000403 BR 2# ;EXIT
3721 020012 004767 176620 1#: JSR PC,WAITIN ;WAIT FOR INTERRUPT
3722 020016 012603 MOV (SP)+,R3 ;STORE REASON POINTER FOR RETURN
3723 020020 2#: ERRHRD 10002...ERR1
      020020 104456 TRAP C#ERHRD
      020022 023422 .WORD 10002
      020024 000000 .WORD 0
      020026 012464 .WORD ERR1
3724 020030 005067 162762 14#: CLR ERRSWI ;CLEAR FOR ERROR RETURN
3725 020034 005767 163066 3#: TST TEMP4 ;TEST IF REGISTERS WERE SAVED
3726 020040 001007 BNE 22# ;NO - SKIP
3727 020042 012703 003034 MOV #L.CS,R3 ;SET POINTER TO RESTORE
3728 020046 012701 000004 MOV #4,R1 ;SET REGISTER COUNT
3729 020052 012623 20#: MOV (SP)+,(R3)+ ;RESTORE REG
3730 020054 005301 DEC R1 ;DEC COUNT
3731 020056 001375 BNE 20# ;LOOP UNTIL ALL ARE RESTORED
3732 020060 162767 000002 162714 22#: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUTINE STACK
3733 020066 012601 MOV (SP)+,R1 ;RESTORE R1
3734 020070 012600 MOV (SP)+,R0 ;RESTORE R0
3735 020072 012603 MOV (SP)+,R3 ;RESTORE R3
3736 020074 012667 163026 MOV (SP)+,TEMP4 ;RESTORE TEMP4
3737 020100 005767 162712 TST ERRSWI ;TEST IF ERROR RETURN
3738 020104 001404 BEQ 99# ;YES - SKIP
3739 020106 066716 162704 ADD ERRSWI,(SP) ;ADD IN FOR NO ERROR RETURN
3740 020112 000240 NOP
3741 020114 000207 RTS PC
3742 020116 017616 000000 99#: MOV #B(SP),(SP) ;SET ERROR RETURN ADDRESS
3743 020122 000240 NOP
3744 020124 000207 RTS PC
3745
3746 ;
3747 020126 010346 ; GDRSTA: GET DRIVE STATE ROUTINE
3748 020130 012701 000004 MOV R3,-(SP) ;SAVE R3
3749 020134 012703 003044 MOV #4,R1 ;INITIALIZE REGISTER SAVE COUNT
3750 020140 014346 1#: MOV #L.MP+2,R3 ;INITIALIZE ADDRESS OF FIRST SAVE
3751 020142 005301 MOV -(R3),-(SP) ;SAVE REGISTER ON STACK
3752 020144 001375 DEC R1 ;DECREMENT REGISTER SAVE COUNT
3753 020146 012767 000003 162664 BNE 1# ;LOOP UNTIL ALL 4 REGISTERS ARE SAVED
3754 MOV #GETSTAT,L.DA ;SET UP DISK ADDRESS REGISTER FOR GET STATUS
      ;/COMMAND
3755 020154 005067 162626 CLR DONE ;CLEAR INTERRUPT FLAG
3756 020160 016767 162646 162646 MOV RLDRV,L.CS ;SET UP CONTROL STATUS REGISTER WITH
      ;/DRIVE NUMBER
3757 BIC #BIT10,L.CS ;CLEAR FOR DRIVES 4-7 SPECIFIED
3758 020166 042767 002000 162640 BIS #GTSTAT,L.CS ;INITIALIZE CONTROL STATUS REGISTER FOR
      ;/GET STATUS COMMAND
3759 020174 052767 000104 162632 MOV L.DA,RLDA(R2) ;INITIALIZE DISK ADDRESS REGISTER FOR
      ;/GET STATUS COMMAND
3760 MOV L.CS,RLCSR(R2) ;LOAD CONTROL STATUS REGISTER TO EXECUTE
3761 020202 016762 162632 000004
3762
3763 020210 016762 162620 000000

```

GLOBAL SUBROUTINES

```

3764
3765 020216 105762 000000      5$: TSTB      RLCS(R2)      ;/GET STATUS COMMAND
3766 020222 001775              BEQ          5$      ;WAIT FOR CONTROLLER READY INDICATING
3767 020224 005767 162556      TST          DONE     ;/RECEIPT OF GET STATUS COMMAND
3768 020230 001416              BEQ          3$      ;INTERRUPT OCCURRED?
3769 020232 016767 162614 162620 MOV          T.MP,T.STAT ;BRANCH IF NOT
3770 020240 042767 177770 162612 BIC          #+C<STAMSK>,T.STAT ;GET CONTENTS OF MULTI-PURPOSE REGISTER
3771 020246 012703 003034      MOV          #L.CS,R3 ;CLEAR ALL BUT STATE DRIVE BITS
3772 020252 012701 000004      MOV          #4,R1    ;INITIALIZE POINTER TO RESTORE RL REGISTERS
3773 020256 012623              MOV          (SP)+,(R3)+ ;INITIALIZE REGISTER SAVE COUNT
3774 020260 005301              DEC          R1       ;RESTORE REGISTERS
3775 020262 001375              BNE         2$      ;DECREMENT REGISTER SAVE COUNT
3776 020264 000402              BR          4$      ;LOOP UNTIL ALL 4 REGISTERS ARE RESTORED
3777 020266 004767 176344      3$: JSR      PC,WAITIN ;WAIT FOR INTERRUPT
3778 020272 012603              4$: MOV      (SP)+,R3  ;RESTORE R3
3779 020274 000207              RTS      PC         ;RETURN
3780
3781 ; SEEK ROUTINE
3782 020276 012767 177777 162614 XSEEK: MOV    #-1,TEMP1 ;SET SPECIAL TIMING SEEK FLAG
3783 020304 000402              BR      XSEEK1
3784 020306 005067 162606      XSEEK: CLR    TEMP1    ;CLEAR SPECIAL TIMING SEEK FLAG
3785 020312 010346              XSEEK1: MOV   R3,-(SP) ;STORE R3
3786 020314 016703 162462      MOV   SSINDX,R3    ;GET SUBROUTINE INDEX
3787 020320 005723              TST    (R3)+       ;BUMP IT FOR NEXT ENTRY
3788 020322 016663 000002 002404 MOV   2(SP),SUBSTK(R3) ;INSERT THIS CALL
3789 020330 162763 000004 002404 SUB   #4,SUBSTK(R3)  ;ADJUST IT TO CALLING LOCATION
3790 020336 010367 162440      MOV   R3,SSINDX    ;STORE IT BACK
3791 020342 010046              MOV   R0,-(SP)
3792 020344 010146              MOV   R1,-(SP)
3793 020346 010546              MOV   R5,-(SP)     ;STORE REG
3794 020350 012767 000002 162440 MOV   #2,ERRSWI    ;SET FOR NO ERROR RETURN
3795 020356 005067 162514      CLR    DIFAUG      ;CLEAR DIFFERENCE ARGUMENT (FOR SEEKING
3796 ; PAST GUARD BAND)
3797 020362 004767 002530              JSR    PC,GETPOS   ;GET PRESENT POSITION
3798 020366 021036              65$
3799 020370 016767 162510 162502 MOV   CURCYL,OLDCYL ;MOVE CURRENT TO OLD CYLINDER
3800 020376 026767 162500 161676 CMP   NEWCYL,HLMTW  ;TEST IF NEW IS GREATER THAN 255
3801 020404 003427              BLE    3$         ;NO - SKIP
3802 020406 166767 161670 162466 SUB   HLMTW,NEWCYL  ;ELSE SUBTRACT 255.
3803 020414 016767 162462 162454 MOV   NEWCYL,DIFAUG ;STORE DIFFERENCE AS ARGUMENT
3804 020422 016767 161654 162452 MOV   HLMTW,NEWCYL ;SET NEWCYL AS 255.
3805 020430 022767 000001 161640 CMP   #1,T.DRIVE
3806 020436 001424              BEQ   6$
3807 020440 162767 000001 162434 SUB   #1,NEWCYL
3808 020446 012767 000001 162434 MOV   #1,DESSGN
3809 020454 012767 000001 162424 MOV   #1,DESDIF
3810 020462 000451              BR    18$
3811 020464 005767 162412      3$: TST    NEWCYL    ;TEST IF NEWCYL HAS NEGATIVE VALUE
3812 020470 100007              BPL    6$        ;NO - SKIP
3813 020472 005467 162404      NEG   NEWCYL      ;ELSE MAKE IT POSITIVE
3814 020476 016767 162400 162372 MOV   NEWCYL,DIFAUG ;AND STORE IT AS ARGUMENT
3815 020504 005067 162372      CLR   NEWCYL      ;AND SET NEWCYL TO 0
3816 020510 016705 162370      6$: MOV   CURCYL,R5  ;COMPUTE DIFFERENCE AND NEW CYLINDER
3817 020514 166705 162362      SUB   NEWCYL,R5   ;SUB NEWCYL FROM CURCYL
3818 020520 100005              BPL    13$       ;IF DIFF IS POSITIVE - SKIP(REV SEEK)
3819 020522 012767 000001 162360 MOV   #1,DESSGN   ;ELSE SET SIGN FOR FORWARD
3820 020530 005405              NEG   R5         ;MAKE DIFFERENCE POSITIVE

```

GLOBAL SUBROUTINES

3821	020532	000402			BR	14#	;SKIP
3822	020534	005067	162350		CLR	DESSGN	;SET SIGN FOR REVERSE
3823	020540	010567	162342	13#:	MOV	R5,DESDIF	;STORE DIFFERENCE
3824	020544	005767	162326	14#:	TST	DIFAUG	;IS THERE A DIFFERENCE ARGUMENT
3825	020550	001416			BEQ	18#	;NO - SKIP
3826	020552	026767	162324	161522	CMP	NEWCYL,HLMTW	;CHECK IF NEW CYL IS 255.
3827	020560	001007			BNE	17#	;NO - SKIP
3828	020562	012767	000001	162320	MOV	#1,DESSGN	;ELSE FORCE SIGN FOR FORWARD
3829							; (INNER GUARD BAND)
3830	020570	022767	000001	161500	CMP	#1,T.DRIVE	
3831	020576	001003			BNE	18#	
3832	020600	066767	162272	162300	ADD	DIFAUG,DESDIF	
3833	020606						
3834	020606	012705	003034		MOV	#L,CS,R5	;GET RL REG ADDRESS
3835	020612	012715	000106		MOV	#SEEK,(R5)	;SET FOR SEEK
3836	020616	056715	162210		BIS	RLDRV,(R5)	;INSERT DRIVE NUMBER
3837	020622	042725	002000		BIC	#BIT10,(R5)+	;CLEAR IF DRIVE 4 - 7 SPEC'D
3838	020626	005025			CLR	(R5)+	;CLEAR BUS ADDRESS
3839	020630	016715	162252		MOV	DESDIF,(R5)	;LOAD DIFFERENCE
3840	020634	012700	000007		MOV	#7,R0	;SET TO SHIFT DIFFERENCE
3841	020640	006315			ASL	(R5)	
3842	020642	005300			DEC	R0	
3843	020644	001375			BNE	21#	;LOOP UNTIL ALIGNED
3844	020646	005767	162236		TST	DESSGN	;TEST SIGN
3845	020652	001402			BEQ	23#	;SKIP IF 0
3846	020654	052715	000004		BIS	#DIRBIT,(R5)	;ELSE INSERT SIGN
3847	020660	005767	162226		TST	DESHD	;TEST IF HEAD 0
3848	020664	001402			BEQ	25#	;YES - SKIP
3849	020666	052715	000020		BIS	#HSEL,(R5)	;ELSE SET HEAD BIT
3850	020672	052725	000001		BIS	#MSET0,(R5)+	;INSERT MARKER BIT
3851	020676	004767	000504		JSR	PC,RDYCHK	;CHECK IF DRIVE READY
3852	020702	021036				65#	
3853	020704	005067	162076		CLR	DONE	;CLEAR INTERRUPT FLAG
3854	020710	005767	162204		TST	TEMP1	;CHECK IF SPECIAL SEEK FLAG SET
3855	020714	001050			BNE	65#	;YES - SKIP, DO NOT START SEEK
3856	020716	014562	000004		MOV	-(R5),RLDA(R2)	;LOAD RL REGISTERS
3857	020722	014562	000002		MOV	-(R5),RLBA(R2)	
3858	020726	014562	000000		MOV	-(R5),RLCS(R2)	;PERFORM SEEK OPERATION
3859					WAITUS	1	;ALLOW TIME FOR RECEIPT OF SEEK COMMAND ;JSD REV A
3860	020732				WAITUS	8.	;ALLOW TIME FOR RECEIPT OF SEEK COMMAND ;JSD REV A
3861	020762	005767	162020		TST	DONE	;TEST IF INTERRUPT DONE
3862	020766	001012			BNE	32#	;YES - SKIP
3863	020770	004767	175642		JSR	PC,WAITIN	;GO WAIT FOR INTERRUPT
3864	020774	012603			MOV	(SP),R3	;GET RESULT MESSAGE POINTER
3865	020776				ERRHRD	10005...,ERR1	
	020776	104456			TRAP	C#ERRHRD	
	021000	023425			.WORD	10005	
	021002	000000			.WORD	0	
	021004	012464			.WORD	ERR1	
3866	021006	005067	162004		CLR	ERRSWI	;CLEAR FOR ERROR RETURN
3867	021012	000411			BR	65#	
3868	021014	005767	162024		TST	T,CS	;TEST IF ANY ERROR
3869	021020	100006			BPL	65#	;NO - SKIP
3870	021022				ERRHRD	10006...,ERR6	
	021022	104456			TRAP	C#ERRHRD	
	021024	023426			.WORD	10006	
	021026	000000			.WORD	0	

GLOBAL SUBROUTINES

Address	Offset	Address	Label	Word	Operation	Comment
3871	021030	012766		ERR6	CLR	ERRSWI ;CLEAR FOR ERROR RETURN
3872	021032	005067	161760		SUB	#2,SSINDX ;REMOVE ENTRY FROM SUBROUTINE STACK
3873	021036	162767	000002	161736	65:	MOV (SP)+,R5 ;RESTORE REGISTER
3874	021044	012605			MOV (SP)+,R1	
3875	021046	012601			MOV (SP)+,R0	
3876	021050	012600			MOV (SP)+,R3 ;RESTORE R3	
3877	021052	012603			TST ERRSWI ;TEST IF ERROR RETURN	
3878	021054	005767	161736		BEQ 99: ;YES - SKIP	
3879	021060	001403			ADD ERRSWI,(SP) ;ADD IN ERROR RETURN	
3880	021062	066716	161730		RTS PC	
3881	021066	000207			MOV 99:(SP),(SP) ;SET ERROR RETURN ADDRESS	
3882	021070	017616	000000	99:	RTS PC	
3883	021074	000207				
3885						
3886	021076	010346		SIMSEK:	MOV R3,-(SP) ;STORE REGISTERS	
3887	021100	016703	161676		MOV SSINDX,R3 ;GET SUBROUTINE INDEX	
3888	021104	005723			TST (R3)+ ;BUMP IT FOR NEXT ENTRY	
3889	021106	016663	000002	002404	MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL	
3890	021114	162763	000004	002404	SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION	
3891	021122	010367	161654		MOV R3,SSINDX ;STORE IT BACK	
3892	021126	010046			MOV R0,-(SP)	
3893	021130	010446			MOV R4,-(SP)	
3894	021132	012767	000002	161656	MOV #2,ERRSWI ;SET FOR NO ERROR RETURN	
3895	021140	004767	000242		JSR PC,RDYCHK ;CHECK IF DRIVE READY	
3896	021144	021350			65:	
3897	021146	012704	003034		MOV #L,CS,R4 ;GET POINTER TO L REGS	
3898	021152	012714	000106		MOV #SEEK,(R4) ;SET FOR SEEK	
3899	021156	056714	161650		BIS RLDRV,(R4) ;INSERT DRIVE NUMBER	
3900	021162	042724	002000		BIC #BIT10,(R4)+ ;CLEAR FOR DRIVE 4 - 7 SPEC'D	
3901	021166	005024			CLR (R4)+ ;CLEAR BUS ADDRESS	
3902	021170	016714	161712		MOV DESDIF,(R4) ;LOAD DIFFERENCE	
3903	021174	012703	000007		MOV #7,R3 ;SET COUNT FOR SHIFT TO ALIGN	
3904	021200	006314		3:	ASL (R4) ;ALIGN DIFFERENCE IN DA	
3905	021202	005303			DEC R3	
3906	021204	001375			BNE 3: ;TEST IF SIGN SET	
3907	021206	005767	161676		TST DESSGN ;NO - SKIP	
3908	021212	001402			BEQ 5: ;INSERT SIGN	
3909	021214	052714	000004		BIS #DIRBIT,(R4) ;TEST IF HEAD 0	
3910	021220	005767	161666	5:	TST DESHD ;YES - SKIP	
3911	021224	001402			BEQ 7: ;INSERT HEAD BIT	
3912	021226	052714	000020		BIS #HSEL,(R4) ;INSERT MARKER BIT	
3913	021232	052724	000001	7:	BIS #MSET0,(R4)+ ;CLEAR INTERRUPT FLAG	
3914	021236	005067	161544		CLR DONE ;SET WAIT COUNT FOR 800US	
3915	021242	012701	000012		MOV #10,R1 ;LOAD RL REGISTERS	
3916	021246	014462	000004		MOV -(R4),RLDA(R2)	
3917	021252	014462	000002		MOV -(R4),RLBA(R2)	
3918	021256	014462	000000		MOV -(R4),RLCS(R2)	
3919	021262	005767	161520	10:	TST DONE ;CHECK IF INTERRUPTED	
3920	021266	001030			BNE 65: ;YES - SKIP	
3921	021270	005301			DEC R1 ;DEC WAIT COUNT	
3922	021272	001415			BEQ 13: ;IF 0 - SKIP	
3923	021274				WAITUS 1	
3924	021324	000756			BR 10: ;GO CHECK DONE	
3925	021326	004767	175304	13:	JSR PC,WAITIN ;GO WAIT FOR TIMEOUT	
3926	021332	012603			MOV (SP)+,R3 ;GET RESULT MESSAGE POINTER	
3927	021334				ERRHRD 10011,ERR1	

GLOBAL SUBROUTINES

```

021334 104456 TRAP C:ERHRD
021336 023433 .WORD 10011
021340 000000 .WORD 0
021342 012464 .WORD ERR1
3928 021344 005067 161446 CLR ERRSWI ;CLEAR FOR ERROR RETURN
3929 021350
3930 021350 162767 000002 161424 14: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
3931 021356 012604 MOV (SP)+,R4 ;RESTORE REGS
3932 021360 012600 MOV (SP)+,R0
3933 021362 012603 MOV (SP)+,R3
3934 021364 005767 161426 TST ERRSWI ;TEST IF ERROR RETURN
3935 021370 001403 BEQ 99: ;YES - SKIP
3936 021372 066716 161420 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
3937 021376 000207 RTS PC
3938 021400 017616 000000 99: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
3939 021404 000207 RTS PC
3941
4017
4018 ;
4019 ; DRIVE READY TEST ROUTINE. CHECKS DRIVE IS READY. IF NOT, WAIT
4020 021406 010346 RDYCHK: MOV R3,-(SP) ;STORE REGS
4021 021410 016703 161366 MOV SSINDX,R3 ;GET SUBROUTINE INDEX
4022 021414 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
4023 021416 016663 000002 002404 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
4024 021424 162763 000004 002404 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
4025 021432 010367 161344 MOV R3,SSINDX ;STORE IT BACK
4026 021436 010046 MOV R0,-(SP)
4027 021440 010146 MOV R1,-(SP)
4028 021442 010446 MOV R4,-(SP)
4029 021444 012767 000002 161344 MOV #2,ERRSWI ;SET FOR NO ERROR RETURN
4030 021452 012701 011610 MOV #5000,R1 ;SET WAIT COUNT
4031 021456 004767 175376 1: JSR PC,GSTAT ;GET DRIVE STATUS
4032 021462 021666
4033 021464 032767 000001 161352 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
4034 021472 001077 BNE 5: ;YES - EXIT
4035 021474 WAITUS 1
4036 021524 005301 DEC R1 ;DEC WAIT COUNT
4037 021526 001353 BNE 1: ;LOOP IF NOT 0
4038 021530 012703 010404 MOV #MDRDY,R3 ;SET RESULT MESSAGE POINTER
4039 021534 012704 011404 MOV #C500MS,R4 ;SET CONDITION MESSAGE POINTER
4040 021540
021540 104456 TRAP C:ERHRD
021542 023432 .WORD 10010
021544 000000 .WORD 0
021546 012716 .WORD ERR5
4041 021550 012701 000030 MOV #24,R1 ;INITIALIZE WAIT COUNT
4042 021554 004767 175300 2: JSR PC,GSTAT ;GET DRIVE STATUS
4043 021560 021666
4044 021562 032767 000001 161254 BIT #DRDYMSK,T.CS ;TEST IF DRIVE READY
4045 021570 001024 BNE 3: ;YES - SKIP
4046 021572 WAITMS 1 ;WAIT FOR 100MS
4047 021636 005301 DEC R1 ;DEC WAIT COUNTER
4048 021640 001345 BNE 2: ;LOOP UNTIL TIME DONE
4049 021642 032767 100000 161174 3: BIT #ANYERR,T.CS ;TEST IF ANYERR SET
4050 021650 001406 BEQ 4: ;NO - SKIP
4051 021652 104456 ERRHRD 10011,,,ERR6 ;REPORT ALL ERRORS
021652 TRAP C:ERHRD

```

GLOBAL SUBROUTINES

```

021654 023433 .WORD 10011
021656 000000 .WORD 0
021660 012766 .WORD ERR6
4052 021662 005367 161272 DEC ERRCNT ;REDUCE ERROR COUNT FOR DUAL ERRORS
4053 021666 005067 161124 4: CLR ERRSWI ;CLEAR FOR ERROR RETURN
4054 021672 162767 000002 161102 5: SUB #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK
4055 021700 012604 MOV (SP)+,R4 ;RESTORE REGS
4056 021702 012601 MOV (SP)+,R1
4057 021704 012600 MOV (SP)+,R0
4058 021706 012603 MOV (SP)+,R3
4059 021710 005767 161102 TST ERRSWI ;TEST IF ERROR RETURN
4060 021714 001403 BEQ 99: ;YES - SKIP
4061 021716 066716 161074 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
4062 021722 000207 RTS PC
4063 021724 017616 000000 99: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
4064 021730 000207 RTS PC
4065
4066 ; CHOOSE HEAD ROUTINE. PICKS HEAD 0 UNLESS SPECIFIC HEAD IS
4067 ; SELECTED BY SOFTWARE PARAMETER.
4068 021732 005067 161154 CHOSHD: CLR DESHD ;CLEAR TO HEAD 0
4069 021736 032767 010000 172362 BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
4070 021744 001403 BEQ 1: ;NO - SKIP
4071 021746 016767 172362 161136 MOV HEADW,DESHD ;INSERT SPECIFIED HEAD
4072 021754 000207 1: RTS PC
4073
4074 ; SHAP HEAD ROUTINE. CHANGES SELECTED HEAD TO HEAD 1
4075 ; UNLESS HEAD 0 SPECIFICALLY SELECTED BY SOFTWARE PARAMETER.
4076 021756 032767 010000 172342 SWAPHD: BIT #HEADLM,MISWIW ;TEST IF HEAD SPECIFIED
4077 021764 001011 BNE 2: ;YES - TAKE ABORT EXIT
4078 021766 005767 161120 TST DESHD ;TEST IF HEAD ONE USED
4079 021772 001006 BNE 2: ;YES - TAKE ABORT EXIT
4080 021774 012767 000001 161110 MOV #1,DESHD ;ELSE SET FOR HEAD ONE
4081 022002 062716 000002 ADD #2,(SP) ;BUMP PAST ABORT RETURN
4082 022006 000207 RTS PC ;RETURN
4083 022010 017616 000000 2: MOV @ (SP),(SP) ;GET ABORT DESTINATION
4084 022014 000207 3: RTS PC
4085
4086 ; SWAP OLD CYLINDER AND NEW CYLINDER ROUTINE.
4087 022016 010046 ONSWAP: MOV RO,-(SP) ;STORE RO
4088 022020 016700 161054 MOV OLDCYL,RO ;MOVE OLD TO RO
4089 022024 016767 161052 161046 MOV NEWCYL,OLDCYL ;MOVE NEW TO OLD
4090 022032 010067 161044 MOV RO,NEWCYL ;PUT OLD IN NEW
4091 022036 012600 MOV (SP)+,RO ;RESTORE RO
4092 022040 000207 RTS PC
4093
4108
4109 ; READ HEADERS ROUTINE.
4110 022042 012767 000001 161056 XRDHDC: MOV #1,TEMP4 ;SET FLAG TO BYPASS REG STORAGE
4111 022050 000402 BR XRDHDG ;GO DO IT
4112 022052 005067 161050 XRDHD: CLR TEMP4 ;SET FLAG TO SAVE T. AND L. REGS
4113 022056 010346 XRDHDG: MOV R3,-(SP) ;STORE REGISTERS
4114 022060 016703 160716 MOV SSINDEX,R3 ;GET SUBROUTINE INDEX
4115 022064 005723 TST (R3)+ ;BUMP IT FOR NEXT ENTRY
4116 022066 016663 000002 002404 MOV 2(SP),SUBSTK(R3) ;INSERT THIS CALL
4117 022074 162763 000004 002404 SUB #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
4118 022102 010367 160674 MOV R3,SSINDEX ;STORE IT BACK
4119 022106 010046 MOV RO,-(SP)

```

GLOBAL SUBROUTINES

4120	022110	010146			MOV	R1,-(SP)	
4121	022112	010446			MOV	R4,-(SP)	
4122	022114	012767	000002	160674	MOV	#2,ERRSWI	;SET FOR NO ERROR RETURN
4123	022122	005767	161000		TST	TEMP4	;TEST IF REGISTERS TO BE SAVED
4124	022126	001007			BNE	2#	;NO - SKIP
4125	022130	012703	003044		MOV	#L.MP+2,R3	;SET POINTER FOR REGS
4126	022134	012701	000004		MOV	#4,R1	;SET COUNT
4127	022140	014346		1#:	MOV	-(R3),-(SP)	;SAVE REGISTER
4128	022142	005301			DEC	R1	;DEC COUNT
4129	022144	001375			BNE	1#	;LOOP UNTIL ALL ARE SAVED
4130	022146	004767	177234	2#:	JSR	PC,RDYCHK	;CHECK DRIVE READY
4131	022152	022440			65#		
4132	022154	005067	160626		CLR	DONE	;CLEAR INTERRUPT FLAG
4133	022160	012701	003034		MOV	#L.CS,R1	;GET ADDRESS OF LOAD REGS
4134	022164	016711	160642		MOV	RLDRV,(R1)	;LOAD DRIVE NUMBER
4135	022170	042711	002000		BIC	#BIT10,(R1)	;CLEAR FOR DRIVE 4 - 7 SPEC'D
4136	022174	052721	000110		BIS	#RDHEAD,(R1)+	;INSERT COMMAND
4137	022200	005021			CLR	(R1)+	;CLEAR BA
4138	022202	005021			CLR	(R1)+	;CLEAR DA
4139	022204	014162	000004		MOV	-(R1),RLDA(R2)	;LOAD RL11 REGS
4140	022210	014162	000002		MOV	-(R1),RLBA(R2)	
4141	022214	014162	000000		MOV	-(R1),RLCSR(R2)	
4142	022220			3#:	WAITUS	10.	;WAIT 1 MS FOR INTERRUPT
4143	022250	005767	160532		TST	DONE	;TEST IF INTERRUPT FLAG SET
4144	022254	001460			BEQ	14#	;NO - SKIP
4145	022256	032767	000001	160560	5#:	BIT	#DRDYMSK,T.CS
4146	022264	001035			BNE	10#	;TEST IF DRIVE READY
4147	022266	012703	010404		MOV	#DRDY,R3	;YES - SKIP
4148	022272	012704	011423		MOV	#CAFDT,R4	;SET NO READY MESSAGE
4149	022276				ERRHRD	10017...ERR5	;CONDITION OF AFTER DATA XFER
	022276	104456			TRAP	C#ERRHRD	
	022300	023441			.WORD	10017	
	022302	000000			.WORD	0	
	022304	012716			.WORD	ERR5	
4150	022306	012701	000030		MOV	#24,R1	;INITIALIZE WAIT COUNT
4151	022312	004767	174542	4#:	JSR	PC,GSTAT	;GET STATUS
4152	022316	022434			60#		
4153	022320	032767	000001	160516	BIT	#DRDYMSK,T.CS	;TEST IF DRIVE HAS COME READY
4154	022326	001403			BEQ	11#	;NO - SKIP
4155	022330	005067	160462		CLR	ERRSWI	;CLEAR ERROR SWITCH
4156	022334	000411			BR	10#	;SKIP
4157	022336	005301		11#:	DEC	R1	;DEC WAIT COUNT
4158	022340	001364			BNE	4#	;LOOP UNTIL TIME DONE
4159	022342	012704	011434		MOV	#CSSEC,R4	;SET CONDITION AFTER 5 SECONDS
4160	022346				ERRHRD	10014...ERR5	
	022346	104456			TRAP	C#ERRHRD	
	022350	023436			.WORD	10014	
	022352	000000			.WORD	0	
	022354	012716			.WORD	ERR5	
4161	022356	000426			BR	60#	;EXIT
4162	022360	005767	160460	10#:	TST	T.CS	;CHECK FOR ANY ERRORS
4163	022364	100005			BPL	12#	;NO - SKIP
4164	022366				ERRHRD	10016...ERR6	;REPORT ALL ERRORS
	022366	104456			TRAP	C#ERRHRD	
	022370	023440			.WORD	10016	
	022372	000000			.WORD	0	
	022374	012766			.WORD	ERR6	

GLOBAL SUBROUTINES

```

4165 022376 000416          BR      60$
4166 022400 012701 003054    12$:  MOV    #HDWRD2,R1      ;GET POINTER
4167 022404 016221 000006    MOV    RLMP(R2),(R1)+  ;STORE LAST TWO HEADER WORDS
4168 022410 016221 000006    MOV    RLMP(R2),(R1)+
4169 022414 000411          BR      65$
4170 022416 004767 174214    14$:  JSR    PC,WAITIN     ;WAIT FOR INTERRUPT
4171 022422 012603          MOV    (SP)+,R3       ;GET RESULTS
4172 022424          ERRHRD 10015,,ERR1  ;REPORT
      022424 104456          TRAP  C#ERRHRD
      022426 023437          .WORD 10015
      022430 000000          .WORD 0
      022432 012464          .WORD ERR1
4173 022434 005067 160356    60$:  CLR    ERRSWI         ;CLEAR FOR ERROR RETURN
4174 022440 005767 160462    65$:  TST    TEMP4         ;TEST IF REGISTERS WERE SAVED
4175 022444 001007          BNE    22$           ;NO - SKIP
4176 022446 012703 003034    MOV    #L.CS,R3      ;SET POINTER TO RESTORE REGS
4177 022452 012701 000004    MOV    #4,R1         ;SET COUNT
4178 022456 012623          20$:  MOV    (SP)+,(R3)+   ;RESTORE REGISTER
4179 022460 005301          DEC    R1            ;DEC COUNT
4180 022462 001375          BNE    20$          ;LOOP UNTIL ALL ARE RESTORED
4181 022464 162767 000002 160310 22$:  SUB    #2,SSINDEX   ;REMOVE ENTRY FROM SUBROUT STACK
4182 022472 012604          MOV    (SP)+,R4
4183 022474 012601          MOV    (SP)+,R1
4184 022476 012600          MOV    (SP)+,R0
4185 022500 012603          MOV    (SP)+,R3
4186 022502 005767 160310    TST    ERRSWI         ;TEST IF ERROR RETURN
4187 022506 001403          BEQ    99$          ;YES - SKIP
4188 022510 066716 160302    ADD    ERRSWI,(SP)   ;ADD IN ERROR RETURN
4189 022514 000207          RTS    PC
4190 022516 017616 000000    99$:  MOV    @B(SP),(SP)   ;SET ERROR RETURN ADDRESS
4191 022522 000207          RTS    PC
4192
4268
4269
      ; POSITION HEAD BIT FROM HEADER OR MULTIPURPOSE REGISTER TO LSB.
4270 022524 016705 160322    POSHM1: MOV   HDWRD1,R5   ;START FOR POSITION HD BIT IN WD 1
4271 022530 000402          BR      POSMDO
4272 022532 016705 160314    POSHSB: MOV   T.MP,R5   ;START FOR POSITION HD BIT IN MP
4273 022536 010146    POSMDO: MOV   R1,-(SP)   ;STORE R1
4274 022540 042705 177677    BIC    #CHSSTAT,R5   ;CLEAR ALL BUT HEAD SEL BIT
4275 022544 012701 000006    MOV    #6,R1         ;SET SHIFT COUNT
4276 022550 006205          1$:  ASR    R5            ;SHIFT FOR RIGHT JUSTIFY
4277 022552 005301          DEC    R1
4278 022554 001375          BNE    1$
4279 022556 012601          MOV    (SP)+,R1     ;RESTORE R1
4280 022560 000207          RTS    PC           ;RETURN
4281
4282
      ; WAIT FOR READY ROUTINE. DURATION OF WAIT PASSED TO THE ROUTINE
      ; FROM THE CALLING ROUTINE IN R1.
4283
      RDYWAIT: MOV   R3,-(SP)   ;STORE R3
4284 022562 010346          MOV    SSINDEX,R3   ;GET SUBROUTINE INDEX
4285 022564 016703 160212    TST    (R3)+        ;BUMP IT FOR NEXT ENTRY
4286 022570 005723          MOV    2(SP),SUBSTK(R3) ;INSERT THIS CALL
4287 022572 016663 000002 002404  SUB    #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
4288 022600 162763 000004 002404  MOV    R3,SSINDEX   ;STORE IT BACK
4289 022606 010367 160170    MOV    R0,-(SP)
4290 022612 010046          MOV    R1,-(SP)
4291 022614 010146          MOV    R4,-(SP)
4292 022616 010446          MOV

```

GLOBAL SUBROUTINES

```

4293 022620 012767 000002 160170      MOV    #2,ERRSWI      ;SET FOR NO ERROR RETURN
4294 022626 004767 174226      5#:   JSR    PC,GSTAT    ;GET DRIVE STATUS
4295 022632 023052                10#
4296 022634 032767 000001 160202  BIT    #DRDYMSK,T.CS ;CHECK IF READY
4297 022642 001105                BNE    9#             ;YES - SKIP
4298 022644 005301                DEC    R1             ;DEC WAIT COUNT
4299 022646 001415                BEQ    7#             ;SKIP IF 0
4300 022650                WAITUS 1
4301 022700 000752                BR     5#
4302 022702 012703 010404      7#:   MOV    #MDRDY,R3    ;SET NAME MESSAGE PTR
4303 022706                ERRHRD 10020,,,ERR3  ;REPORT READY ERROR
         022706 104456                TRAP  C#ERRHRD
         022710 023444                .WORD 10020
         022712 000000                .WORD 0
         022714 012600                .WORD ERR3
4304 022716 012701 000030      MOV    #24,,R1       ;INITIALIZE WAIT COUNT
4305 022722 004767 174132      6#:   JSR    PC,GSTAT    ;GET DRIVE STATUS
         10#
4306 022726 023052                BIT    #DRDYMSK,T.CS ;TEST IF DRIVE READY
4307 022730 032767 000001 160106  BNE    8#             ;YES - SKIP
4308 022736 001033                WAITMS 1            ;WAIT 100 MS
4309 022740                DEC    R1             ;DEC WAIT COUNT
4310 023004 005301                BNE    6#             ;LOOP UNTIL TIME DONE
4311 023006 001345                MOV    #C5SEC,R4     ;SET CONDITION AFTER 5 SECDS
4312 023010 012704 011434      ERRHRD 10021,,,ERR5
4313 023014                TRAP  C#ERRHRD
         023014 104456                .WORD 10021
         023016 023445                .WORD 0
         023020 000000                .WORD ERR5
         023022 012716                BR     11#
4314 023024 000410                BIT    #ANYERR,T.CS ;EXIT
4315 023026 032767 100000 160010  8#:   BEQ    10#           ;TEST IF ANY ERROR SET
4316 023034 001406                ERRHRD 10022,,,ERR6 ;NO - SKIP
4317 023036                TRAP  C#ERRHRD     ;REPORT ALL ERRORS
         023036 104456                .WORD 10022
         023040 023446                .WORD 0
         023042 000000                .WORD ERR6
         023044 012766                DEC    ERRCNT        ;DECREMENT FOR DOUBLE ERROR REPORT
4318 023046 005367 160106      11#:  CLR    ERRSWI        ;CLEAR FOR ERROR ERROR RETURN
4319 023052 005067 157740      10#:  SUB    #2,SSINDEX   ;REMOVE ENTRY FROM SUBROUT STACK
4320 023056 162767 000002 157716  9#:   MOV    (SP)+,R4     ;RESTORE REGISTERS
4321 023064 012604                MOV    (SP)+,R1
4322 023066 012601                MOV    (SP)+,R0
4323 023070 012600                MOV    (SP)+,R3     ;RESTORE R3
4324 023072 012603                TST   ERRSWI        ;TEST IF ERROR RETURN
4325 023074 005767 157716                BEQ    99#           ;YES - SKIP
4326 023100 001403                ADD   ERRSWI,(SP)   ;ADD IN ERROR RETURN
4327 023102 066716 157710                RTS    PC
4328 023106 000207                MOV    @((SP)),((SP)) ;SET ERROR RETURN ADDRESS
4329 023110 017616 000000      99#:  RTS    PC
4330 023114 000207
4331
4332                ; GET POSITION ROUTINE. READS A HEADER FROM CURRENT CYLINDER
4333                ; (WHERE IT IS PRESENTLY POSITIONED) AND STORES CYLINDER
4334                ; NUMBER IN CURCYL.
GETPOS: MOV    R3,-(SP)    ;STORE REGISTERS
        MOV    SSINDEX,R3 ;GET SUBROUTINE INDEX
        TST   (R3)+      ;BUMP IT FOR NEXT ENTRY

```

GLOBAL SUBROUTINES

```

4338 023126 016663 000002 002404      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
4339 023134 162763 000004 002404      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
4340 023142 010367 157634      MOV      R3,SSINDEX ;STORE IT BACK
4341 023146 010046      MOV      R0,-(SP)
4342 023150 010546      MOV      R5,-(SP)
4343 023152 004767 176674      JSR      PC,XRDHD ;DO READ HEADER
4344 023156 023206      65#
4345 023160 016703 157666      MOV      HDWRD1,R3 ;GET HEADER WORD
4346 023164 012705 000007      MOV      #7,R5 ;SET SHIFT COUNT
4347 023170 006203      4# :    ASR      R3 ;SHIFT TO RIGHT JUSTIFY
4348 023172 005305      DEC      R5
4349 023174 001375      BNE      4#
4350 023176 042703 177000      BIC      #177000,R3
4351 023202 010367 157676      MOV      R3,CURCYL ;STORE AS CURRENT CYLINDER
4352 023206 162767 000002 157566 65# :    SUB      #2,SSINDEX ;REMOVE ENTRY FROM SUBROUT STACK
4353 023214 012605      MOV      (SP)+,R5 ;RESTORE REGISTERS
4354 023216 012600      MOV      (SP)+,R0
4355 023220 012603      MOV      (SP)+,R3
4356 023222 005767 157570      TST      ERRSWI ;TEST IF ERROR RETURN
4357 023226 001403      BEQ      99# ;YES - SKIP
4358 023230 066716 157562      ADD      ERRSWI,(SP) ;ADD IN ERROR RETURN
4359 023234 000207      RTS      PC
4360 023236 017616 000000      99# :    MOV      #8(SP),(SP) ;SET ERROR RETURN ADDRESS
4361 023242 000207      RTS      PC
4362
4391
4392      ;
4393      ; READ ALL HEADERS ROUTINE. 40 HEADERS ARE READ AND STORED
4394 023244 010346      ;RDALHD: MOV      R3,-(SP) ;STORE REGISTERS
4395 023246 016703 157530      MOV      SSINDEX,R3 ;GET SUBROUTINE INDEX
4396 023252 005723      TST      (R3)+ ;BUMP IT FOR NEXT ENTRY
4397 023254 016663 000002 002404      MOV      2(SP),SUBSTK(R3) ;INSERT THIS CALL
4398 023262 162763 000004 002404      SUB      #4,SUBSTK(R3) ;ADJUST IT TO CALLING LOCATION
4399 023270 010367 157506      MOV      R3,SSINDEX ;STORE IT BACK
4400 023274 010046      MOV      R0,-(SP)
4401 023276 010146      MOV      R1,-(SP)
4402 023300 010446      MOV      R4,-(SP)
4403 023302 012767 000002 157506      MOV      #2,ERRSWI ;SET FOR NO ERROR RETURN
4404 023310 012701 000050      MOV      #40,,R1 ;SET HEADER COUNT
4405 023314 052767 100000 157462      BIS      #HDR40,OPFLAG ;SET 40 HDR OP FLAG
4406 023322 012703 003764      MOV      #IBUFF,R3 ;SET POINTER TO STORE HDRS
4407 023326 016704 157474      MOV      RLBAS,R4 ;GET BASE ADDRESS
4408 023332 062704 000006      ADD      #RLMP,R4 ;MAKE IT POINT TO MP REG
4409 023336 012767 000010 157470      MOV      #10,L.CS ;LOAD FOR READ HEADER, NO INTERRUPT
4410 023344 056767 157462 157462      BIS      RLDRV,L.CS ;INSERT DRIVE NUMBER
4411 023352 042767 002000 157454      BIC      #BIT10,L.CS ;CLEAR FOR DRIVE 4 - 7 SPEC'D
4412 023360 005067 157452      CLR      L.BA ;CLEAR BA
4413 023364 005067 157450      CLR      L.DA ;CLEAR DA
4414 023370 005767 157516      TST      DESHD ;TEST IF HEAD 0
4415 023374 001403      BEQ      3# ;YES - SKIP
4416 023376 052767 000020 157434      BIS      #HDSSEL,L.DA ;ELSE INSERT HEAD 0
4417 023404 016762 157430 000004 3# :    MOV      L.DA,RLDA(R2) ;LOAD RLDA REG
4418 023412 016762 157420 000002      MOV      L.BA,RLBA(R2) ;LOAD RLBA
4419 023420 032762 000200 000000      BIT      #CRDYMSK,RLCS(R2) ;TEST IF CONTROLLER READY
4420 023426 001003      BNE      6# ;YES - SKIP
4421 023430 004767 175752      JSR      PC,RDYCHK ;ELSE CHECK READY
4422 023434 023552      65#

```

GLOBAL SUBROUTINES

```

4423 023436 016762 157372 000000 6#: MOV L.CS,RLCS(R2) ;LOAD RLCS REG
4424 023444 012700 077777 MOV #77777,R0 ;SET COUNT FOR WAIT
4425 023450 032762 000200 000000 7#: BIT #CRDYMSK,RLCS(R2) ;CHECK THAT OPERATION COMPLETED
4426 023456 001016 BNE 8# ;YES - SKIP
4427 023460 005300 DEC R0 ;DEC COUNT
4428 023462 001372 BNE 7# ;SKIP IF NOT YET 0
4429 023464 004767 173114 JSR PC,READRL ;ELSE GET ALL REGISTERS
4430 023470 004767 173142 JSR PC,WAITIN ;ELSE WAIT FOR TIMEOUT
4431 023474 012603 MOV (SP)+,R3 ;GET RESULT MESSAGE POINTER
4432 023476 ERRHRD 10025...,ERR1
023476 104456 TRAP C#ERRHRD
023500 023451 .WORD 10025
023502 000000 .WORD 0
023504 012464 .WORD ERR1
4433 023506 005067 157304 CLR ERRSWI ;CLEAR FOR ERROR RETURN
4434 023512 000417 BR 65#
4435 023514 005767 157324 8#: TST T.CS ;TEST FOR ANY ERRORS
4436 023520 100007 BPL 12# ;NO - SKIP
4437 023522 ERRHRD 10026...,ERR6
023522 104456 TRAP C#ERRHRD
023524 023452 .WORD 10026
023526 000000 .WORD 0
023530 012766 .WORD ERR6
4438 023532 005067 157260 CLR ERRSWI ;CLEAR FOR ERROR RETURN
4439 023536 000405 BR 65#
4440 023540 011423 12#: MOV (R4),(R3)+ ;STORE HEADER WORDS
4441 023542 011423 MOV (R4),(R3)+
4442 023544 011423 MOV (R4),(R3)+
4443 023546 005301 DEC R1 ;DEC HEADER COUNT
4444 023550 001332 BNE 6#
4445 023552 162767 000002 157222 65#: SUB #2,SSINDX ;REMOVE ENTRY FROM SUBROUT STACK
4446 023560 012604 MOV (SP)+,R4 ;RESTORE REGISTERS
4447 023562 012601 MOV (SP)+,R1
4448 023564 012600 MOV (SP)+,R0
4449 023566 012603 MOV (SP)+,R3
4450 023570 005767 157222 TST ERRSWI ;TEST IF ERROR RETURN
4451 023574 001403 BEQ 99# ;YES - SKIP
4452 023576 066716 157214 ADD ERRSWI,(SP) ;ADD IN ERROR RETURN
4453 023602 000207 RTS PC
4454 023604 017616 000000 99#: MOV @ (SP),(SP) ;SET ERROR RETURN ADDRESS
4455 023610 000207 RTS PC

```

```

4456
4684
4685 ; REPORT OPERATION ROUTINE. PRINTS SUBROUTINE TRACE SEQUENCE AND
4686 ; OPERATION BEING PERFORMED PORTION OF ALL
4687 ; ERROR MESSAGES.
4688 RPTOP: MOV R4,-(SP)
4689 023612 010446 TST SSINDX ;TEST SUBROUTINE INDEX 0
4690 023614 005767 157162 BEQ 1# ;SKIP IF 0
4691 023620 001433 MOV #2,R4 ;SET INDEXER TO FIRST ENTRY
4692 023622 012704 000002 PRINTB #FMT9,#SEQMES ;PRINT "SUBROUTINE CALL SEQ"
023626 012746 010253 MOV #SEQMES,-(SP)
023632 012746 011753 MOV #FMT9,-(SP)
023636 012746 000002 MOV #2,-(SP)
023642 010600 MOV SP,R0
023644 104414 TRAP C#PNTB
023646 062706 000006 ADD #6,SP

```

GLOBAL SUBROUTINES

J7

```

4693 023652          34: PRINTB  #FMT16,SUBSTK(R4)      ;PRINT CALLING LOCATION
      023652 016446 002404      MOV     SUBSTK(R4),-(SP)
      023656 012746 012126      MOV     #FMT16,-(SP)
      023662 012746 000002      MOV     #2,-(SP)
      023666 010600      MOV     SP,R0
      023670 104414      TRAP   C#PNTB
      023672 062706 000006      ADD     #6,SP
4694 023676 062704 000002      ADD     #2,R4      ;BUMP INDEX
4695 023702 020467 157074      CMP     R4,SSINDX  ;CHECK IF ALL PRINTED
4696 023706 003761      BLE     34        ;LOOP IF NOT ALL PRINTED YET
4697 023710          14: PRINTB  #FMT4,ERHEAD,#TSTLAB      ;PRINT ERROR HEADER
      023710 012746 006365      MOV     #TSTLAB,-(SP)
      023714 016746 157072      MOV     ERHEAD,-(SP)
      023720 012746 011556      MOV     #FMT4,-(SP)
      023724 012746 000003      MOV     #3,-(SP)
      023730 010600      MOV     SP,R0
      023732 104414      TRAP   C#PNTB
      023734 062706 000010      ADD     #10,SP
4698 023740 042767 030000 157036      BIC     #SEEKOP!RORWOP,OPFLAG ;CLEAR SK & RD OR WRT FLAG
4699 023746 016701 157062      MOV     L.CS,R1    ;GET COMMAND EXECUTED
4700 023752 042701 177741      BIC     #177741,R1 ;STRIP ALL BUT FUNCTION CODE
4701 023756 022701 000006      CMP     #6,R1     ;TEST IF SEEK OPERATION
4702 023762 001003      BNE     24        ;NO - SKIP
4703 023764 052767 010000 157012      BIS     #SEEKOP,OPFLAG ;ELSE SET SEEK FLAG
4704 023772 022701 000012          24: CMP     #12,R1    ;TEST IF WRITE
4705 023776 001003      BNE     204       ;NO - SKIP
4706 024000 052767 020000 156776      BIS     #RORWOP,OPFLAG ;SET RD OR WRT FLAG
4707 024006 022701 000014          204: CMP     #14,R1   ;TEST IF READ
4708 024012 001003      BNE     224       ;NO - SKIP
4709 024014 052767 020000 156762      BIS     #RORWOP,OPFLAG ;SET RD OR WRT FLAG
4710 024022          224: PRINTB  #FMT1,#MOPER,OPMSGs(R1) ;PRINT OPERATION
      024022 016146 002224      MOV     OPMSGs(R1),-(SP)
      024026 012746 005414      MOV     #MOPER,-(SP)
      024032 012746 011534      MOV     #FMT1,-(SP)
      024036 012746 000003      MOV     #3,-(SP)
      024042 010600      MOV     SP,R0
      024044 104414      TRAP   C#PNTB
      024046 062706 000010      ADD     #10,SP
4711 024052 020127 000004      CMP     R1,#4     ;CHECK IF GET STATUS
4712 024056 001007      BNE     44        ;NO - SKIP
4713 024060 032767 000010 156752      BIT     #DRSET,L.DA ;TEST IF RESET INCLUDED
4714 024066 001403      BEQ     44        ;NO - SKIP
4715 024070 012701 000016      MOV     #16,R1   ;SET TO PRINT WITH RESET
4716 024074 000436      BR     94
4717 024076 032767 007777 156700          44: BIT     #COMPOP,OPFLAG ;TEST IF ANY OTHER OPERATION
4718 024104 001424      BEQ     84        ;NO - SKIP
4719 024106 016704 156672      MOV     OPFLAG,R4 ;SET UP TO DETERMINE WHICH ONE
4720 024112 012701 000020      MOV     #20,R1   ;PRESET THE POINTER
4721 024116 032704 000001          54: BIT     #BIT00,R4  ;CHECK THE BIT
4722 024122 001003      BNE     64        ;IF SET - SKIP
4723 024124 005721      TST     (R1)+    ;BUMP POINTER
4724 024126 006204      ASR     R4
4725 024130 000772      BR     54
4726 024132          64: PRINTB  #FMT2,OPMSGs(R1)
      024132 016146 002224      MOV     OPMSGs(R1),-(SP)
      024136 012746 011550      MOV     #FMT2,-(SP)
      024142 012746 000002      MOV     #2,-(SP)

```


GLOBAL SUBROUTINES

```

024146 010600      MOV      SP,R0
024150 104414      TRAP     C#PNTB
024152 062706 000006  ADD      #6,SP
4727 024156 032767 100000 156620 8#:  BIT      #HDR40,OPFLAG ;TEST IF 40 HEADER OPERATION
4728 024164 001415      BEQ      10# ;NO - SKIP
4729 024166 012701 000050      MOV      #50,R1 ;ELSE PRINT IT
4730 024172      PRINTB  #FMT2,OPMSG(S(R1))
024172 016146 002224      MOV      OPMSG(S(R1)),-(SP)
024176 012746 011550      MOV      #FMT2,-(SP)
024202 012746 000002      MOV      #2,-(SP)
024206 010600      MOV      SP,R0
024210 104414      TRAP     C#PNTB
024212 062706 000006  ADD      #6,SP
4731 024216 000434      BR       15# ;SKIP
4732 024220 032767 010000 156556 10#: BIT      #SEEKOP,OPFLAG ;TEST IF SEEK
4733 024226 001430      BEQ      15# ;NO - SKIP
4734 024230      PRINTB  #FMT13,#FRMWD,OLDCYL,#DIFWD,DESDIF,#SGNWD,DESSGN,#HDWD,DESHD
024230 016746 156656      MOV      DESHD,-(SP)
024234 012746 010214      MOV      #HDWD,-(SP)
024240 016746 156644      MOV      DESSGN,-(SP)
024244 012746 010207      MOV      #SGNWD,-(SP)
024250 016746 156632      MOV      DESDIF,-(SP)
024254 012746 010201      MOV      #DIFWD,-(SP)
024260 016746 156614      MOV      OLDCYL,-(SP)
024264 012746 010232      MOV      #FRMWD,-(SP)
024270 012746 011774      MOV      #FMT13,-(SP)
024274 012746 000011      MOV      #11,-(SP)
024300 010600      MOV      SP,R0
024302 104414      TRAP     C#PNTB
024304 062706 000024  ADD      #24,SP
4735 024310 032767 020000 156466 15#: BIT      #RORWOP,OPFLAG ;TEST IF READ OR WRITE SET
4736 024316 001424      BEQ      17# ;NO - SKIP
4737 024320      PRINTB  #FMT22,#CYLWD,CURCYL,#HDWD,DESHD,#SECWD,DESSEC
024320 016746 156570      MOV      DESSEC,-(SP)
024324 012746 010220      MOV      #SECWD,-(SP)
024330 016746 156556      MOV      DESHD,-(SP)
024334 012746 010214      MOV      #HDWD,-(SP)
024340 016746 156540      MOV      CURCYL,-(SP)
024344 012746 010225      MOV      #CYLWD,-(SP)
024350 012746 012323      MOV      #FMT22,-(SP)
024354 012746 000007      MOV      #7,-(SP)
024360 010600      MOV      SP,R0
024362 104414      TRAP     C#PNTB
024364 062706 000020  ADD      #20,SP
4738 024370 004767 000446 17#: JSR      PC,CLRPARM ;CLEAR PARAM TABLE
4739 024374 012604      MOV      (SP)+,R4 ;RESTORE R4
4740 024376 000207      RTS      PC
4741
4742 ;
4743 ; REPORT REASON ROUTINE
RPTRES: PRINTS REASON PORTION FOR ALL ERROR REPORTS.
024400 010146      MOV      R1,-(SP) ;STORE R1
024402 010346      MOV      R3,-(SP) ;STORE R3
024404 010446      MOV      R4,-(SP) ;STORE R4
024406 012701 003062      MOV      #RESPARM,R1 ;GET START OF PARAM
024412 012103      MOV      (R1)+,R3 ;GET NUMBER OF PARAM
4749 024414      PRINTB  #FMT1.1,#MRSLT,(R1) ;PRINT NAME
024414 011146      MOV      (R1),-(SP)

```

GLOBAL SUBROUTINES

	024416	012746	005423	MOV	#MRSLT,-(SP)	
	024422	012746	011541	MOV	#FMT1.1,-(SP)	
	024426	012746	000003	MOV	#3,-(SP)	
	024432	010600		MOV	SP,R0	
	024434	104414		TRAP	C#PNTB	
	024436	062706	000010	ADD	#10,SP	
4750	024442	021127	011057	CMP	(R1),#MNRST	;TEST IF MESSAGE IS NO DRV STATUS
4751	024446	001453		BEQ	6#	;YES - SKIP REST OF REPORT
4752	024450	012704	011760	MOV	#FMT11,R4	;PRESET FOR FORMAT 11
4753	024454	022127	011052	CMP	(R1)+,#MCYLOC	;CHECK IF REPORTING CYLINDER LOC
4754	024460	001002		BNE	3#	;NO - SKIP
4755	024462	012704	011766	MOV	#FMT12,R4	;ELSE CHANGE TO FORMAT 12
4756	024466	005303		DEC	R3	;DEC PARAM COUNT
4757	024470	001442		BEQ	6#	;IF 0 - EXIT
4758	024472			PRINTB	R4,#RESE3,(R1)+	;REPORT IS VALUE
	024472	012146		MOV	(R1)+,-(SP)	
	024474	012746	011300	MOV	#RESE3,-(SP)	
	024500	010446		MOV	R4,-(SP)	
	024502	012746	000003	MOV	#3,-(SP)	
	024506	010600		MOV	SP,R0	
	024510	104414		TRAP	C#PNTB	
	024512	062706	000010	ADD	#10,SP	
4759	024516			PRINTB	R4,#RESE4,(R1)+	;REPORT SB VALUE
	024516	012146		MOV	(R1)+,-(SP)	
	024520	012746	011304	MOV	#RESE4,-(SP)	
	024524	010446		MOV	R4,-(SP)	
	024526	012746	000003	MOV	#3,-(SP)	
	024532	010600		MOV	SP,R0	
	024534	104414		TRAP	C#PNTB	
	024536	062706	000010	ADD	#10,SP	
4760	024542	162703	000002	SUB	#2,R3	;DEC PARAM COUNT
4761	024546	001413		BEQ	6#	;IF 0 - EXIT
4762	024550			PRINTB	#FMT1,#RESE5,(R1)+	;REPORT CONDITION
	024550	012146		MOV	(R1)+,-(SP)	
	024552	012746	011311	MOV	#RESE5,-(SP)	
	024556	012746	011534	MOV	#FMT1,-(SP)	
	024562	012746	000003	MOV	#3,-(SP)	
	024566	010600		MOV	SP,R0	
	024570	104414		TRAP	C#PNTB	
	024572	062706	000010	ADD	#10,SP	
4763	024576	012604		MOV	(SP)+,R4	;RESTORE REGS
4764	024600	012603		MOV	(SP)+,R3	
4765	024602	012601		MOV	(SP)+,R1	
4766	024604	000207		RTS	PC	;RETURN
4767						
4768						
4769						
4770	024606					
	024606	005046				
	024610	156716	156217			
	024614	012746	006053			
	024620	016746	156202			
	024624	012746	006042			
	024630	012746	011567			
	024634	012746	000005			
	024640	010600				
	024642	104414				

```

; REPORT PHYSICAL ADDRESS OF DEVICE UNDER TEST
; AND ALL REGISTER CONTENTS.
RPTREM: PRINTB #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
CLR -(SP)
BISB RLDRV+1,(SP)
MOV #DRVNAM,-(SP)
MOV RLBAS,-(SP)
MOV #BASADD,-(SP)
MOV #FMT5,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C#PNTB

```

GLOBAL SUBROUTINES

```

4771 024644 062706 000014      ADD      #14,SP
4772 024650      ; REPORT RL11 REGISTERS
024650 012746 010214      PRINTB  #FMT6,#CSNAM,#DANAM,#BANAM,#MPNAM,#CYLWD,#HDWD
024654 012746 010225      MOV      #HDWD,-(SP)
024660 012746 006137      MOV      #CYLWD,-(SP)
024664 012746 006125      MOV      #MPNAM,-(SP)
024670 012746 006132      MOV      #BANAM,-(SP)
024674 012746 006120      MOV      #DANAM,-(SP)
024700 012746 011607      MOV      #CSNAM,-(SP)
024704 012746 000007      MOV      #FMT6,-(SP)
024710 010600      MOV      #7,-(SP)
024712 104414      MOV      SP,RO
024714 062706 000020      TRAP    C#PNTB
4773 024720      ADD      #20,SP
024720 016746 156116      PRINTB  #FMT8,#LAB1,L.CS,L.DA,L.BA,L.MP
024724 016746 156106      MOV      L.MP,-(SP)
024730 016746 156104      MOV      L.BA,-(SP)
024734 016746 156074      MOV      L.DA,-(SP)
024740 012746 006144      MOV      L.CS,-(SP)
024744 012746 011721      MOV      #LAB1,-(SP)
024750 012746 000006      MOV      #FMT8,-(SP)
024754 010600      MOV      #6,-(SP)
024756 104414      MOV      SP,RO
024760 062706 000016      TRAP    C#PNTB
4774 024764      ADD      #16,SP
024764 016746 156122      PRINTB  #FMT7,#LAB2,T.CS,T.DA,T.BA,T.MP,CURCYL,DESHD
024770 016746 156110      MOV      DESHD,-(SP)
024774 016746 156052      MOV      CURCYL,-(SP)
025000 016746 156042      MOV      T.MP,-(SP)
025004 016746 156040      MOV      T.BA,-(SP)
025010 016746 156030      MOV      T.DA,-(SP)
025014 012746 006157      MOV      T.CS,-(SP)
025020 012746 011651      MOV      #LAB2,-(SP)
025024 012746 000010      MOV      #FMT7,-(SP)
025030 010600      MOV      #10,-(SP)
025032 104414      MOV      SP,RO
025034 062706 000022      TRAP    C#PNTB
4775 025040 000207      ADD      #22,SP
4776 025040 000207      RTS      PC
4777      ; CLEAR PARAMETER BLOCK FOR REPORTING
4778 025042 010546      CLRPARM: MOV      R5,-(SP)      ;STORE R5
4779 025044 012701 003062      MOV      #RESPARM,R1      ;GET ADDRESS OF BLOCK
4780 025050 012705 000005      MOV      #5,R5      ;SET COUNT
4781 025054 005021      2#: CLR      (R1)+      ;CLEAR WORD
4782 025056 005305      DEC      R5      ;DEC COUNT
4783 025060 001375      BNE      2#      ;LOOP UNTIL 0
4784 025062 012701 003062      MOV      #RESPARM,R1      ;RESET POINTER
4785 025066 012605      MOV      (SP)+,R5      ;RESTORE R5
4786 025070 000207      RTS      PC
4787
4788 025072      ENDMOD
4789
4790      .TITLE  CNRLIA0 RL01/02 DRIVE TEST 1
4791
4792      ;DISK STATE FUNCTIONS
4793      ;

```

GLOBAL SUBROUTINES

```

4794 ;BITS 0-2 OF THE MULTIPURPOSE REGISTER DURING GET STATUS COMMAND DEFINE THE
4795 ;STATE OF THE DRIVE
4796 ;
4797 ; STATE 0 LOAD STATE
4798 ; STATE 1 SPIN UP
4799 ; STATE 2 BRUSH CYCLE
4800 ; STATE 3 LOAD HEADS
4801 ; STATE 4 SEEK
4802 ; STATE 5 LOCK ON
4803 ; STATE 6 UNLOAD HEADS
4804 ; STATE 7 SPIN DOWN
4805
4806 025072 BGNMOD HRDWTST
4807
4808 .SBTTL *TEST 1 BASIC INTERFACE (PART 1)
4809
4810 025072 BGNTST ;TEST01
4811 ;TEST THAT UNLOAD, COVER OPEN AND WRITE PROTECT START T1::
4812 ;IN THE PROPER STATE.
4813 025072 005767 156262 TST PASNUM ;CHECK IF FIRST PASS
4814 025076 001124 BNE 65# ;EXIT IF NO
4815 025100 005767 167222 TST MISWIW ;CHECK IF MANUAL INTERVENTION
4816 025104 100121 BPL 65# ;NO - EXIT TEST
4817 025106 012767 006373 155676 2# MOV #MISTST,ERHEAD ;LOAD ERR HEADER
4818 ;PROMPT CHK DRV IS UNLDED, COVR OPN, AND
4819 ;WRTE LCKED
4820 025114 PRINTF #FMTOP1,#OPR1,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
4821 025114 005046 CLR -(SP)
4822 025116 156716 155711 BISB RLDRV+1,(SP)
4823 025122 012746 006053 MOV #DRVNAM,-(SP)
4824 025126 016746 155674 MOV RLBAS,-(SP)
4825 025132 012746 006042 MOV #BASADD,-(SP)
4826 025136 012746 010135 MOV #OPR1A,-(SP)
4827 025142 012746 007540 MOV #OPR1,-(SP)
4828 025146 012746 011442 MOV #FMTOP1,-(SP)
4829 025152 012746 000007 MOV #7,-(SP)
4830 025156 010600 MOV SP,R0
4831 025160 104417 TRAP C#PNTF
4832 025162 062706 000020 ADD #20,SP
4833
4834 025166 005067 157172 CLR OBUFF ;CLEAR FOR RESPONSE
4835 025172 GMANIL OPRO02,OBUFF,1,NO
4836 025172 104443 TRAP C#GMAN
4837 025174 000404 BR 10000#
4838 025176 004364 .WORD OBUFF
4839 025200 000120 .WORD T#CODE
4840 025202 007470 .WORD OPRO02
4841 025204 000001 .WORD 1
4842 10000#
4843 025206 005767 157152 TST OBUFF ;TEST RESPONSE YES
4844 025212 001735 BEQ 2# ;YES - SKIP
4845 025214 004767 171572 JSR PC,TSTINT ;INITIALIZE TEST
4846 025220 004767 171604 JSR PC,GSTATR ;GO GET STATUS WITH RESET
4847 025224 025350 65#
4848 025226 032767 000040 155616 BIT #COSTAT,T.MP ;CHECK IF COVER OPEN SET
4849 025234 001006 BNE 7# ;YES - SKIP

```

*TEST 1 BASIC INTERFACE (PART 1)

```

4831 025236 012703 010562      MOV      #MCOSTA,R3      ;SET NAME POINTER
4832 025242      ERRHRD  101...ERR3
      025242 104456      TRAP    C#ERRHD
      025244 000145      .WORD  101
      025246 000000      .WORD  0
      025250 012600      .WORD  ERR3
4833 025252 032767 000010 155572 7# :    BIT      #BHSTAT,T.MP    ;TEST IF BRUSHES HOME
4834 025260 001006      BNE     9#              ;YES - SKIP
4835 025262 012703 010575      MOV      #MBHSTA,R3      ;SET POINTER FOR BRUSH HOME ERROR
4836 025266      ERRHRD  102...ERR3
      025266 104456      TRAP    C#ERRHD
      025270 000146      .WORD  102
      025272 000000      .WORD  0
      025274 012600      .WORD  ERR3
4837 025276 032767 020000 155546 9# :    BIT      #MLSTAT,T.MP    ;TEST IF WRITE LOCK SET
4838 025304 001006      BNE     11#            ;YES - SKIP
4839 025306 012703 010610      MOV      #MMLSTA,R3      ;SET NAME POINTER
4840 025312      ERRHRD  103...ERR3
      025312 104456      TRAP    C#ERRHD
      025314 000147      .WORD  103
      025316 000000      .WORD  0
      025320 012600      .WORD  ERR3
4841 025322 005767 155532      11# :    TST      T.STAT          ;TEST IF STATE ZERO
4842 025326 001405      BEQ     15#            ;YES - SKIP
4843 025330 005003      CLR     R3              ;SET STATE EXPECTED
4844 025332      ERRHRD  104...ERR7
      025332 104456      TRAP    C#ERRHD
      025334 000150      .WORD  104
      025336 000000      .WORD  0
      025340 013666      .WORD  ERR7
4845 025342 004767 171462      15# :    JSR      PC,GSTATR      ;DO DRIVE RESET
4846 025346 025350      65# :
4847 025350      65# :
4848 025350      ENDTST
      025350 104401      L10024: TRAP    C#ETST
4849
4850      .SBTTL *TEST 2      BASIC INTERFACE (PART 2)
4851
4852 025352      BGNTST      ;TEST 2
      025352
4853      ;VERIFY THAT COVER OPEN AND WRITE PROTECT WORK.
4854 025352 005767 156002      TST      PASNUM          ;TEST IF PASS 0
4855 025356 001077      BNE     65#            ;NO - SKIP
4856 025360 005767 166742      TST      MISWIW          ;TEST IF MANUAL INTERVENTION
4857 025364 100074      BPL     65#            ;NO - SKIP
4858 025366 012767 006373 155416      MOV      #MISTST,ERHEAD ;SET ERROR HEADER
4859
4860 025374      2# :      ;PROMPT CLOSE COVER AND RESET WRITE LOCK.
4861 025374      PRINTF  #FMTOP1,#OPR2,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      025374 005046      CLR     -(SP)
      025376 156716 155431      BISB    RLDRV+1,(SP)
      025402 012746 006053      MOV     #DRVNAM,-(SP)
      025406 016746 155414      MOV     RLBAS,-(SP)
      025412 012746 006042      MOV     #BASADD,-(SP)
      025416 012746 010135      MOV     #OPR1A,-(SP)
      025422 012746 007616      MOV     #OPR2,-(SP)

```

*TEST 2 BASIC INTERFACE (PART 2)

```

025426 012746 011442      MOV    #FMTOP1,-(SP)
025432 012746 000007      MOV    #7,-(SP)
025436 010600      MOV    SP,R0
025440 104417      TRAP   C#PNTF
025442 062706 000020      ADD    #20,SP
4862 025446 005067 156712      CLR    OBUFF ;CLEAR FOR RESPONSE
4863 025452      GMANIL OPR002,OBUFF,1,NO
025452 104443      TRAP   C#GMAN
025454 000404      BR     10000#
025456 004364      .WORD OBUFF
025460 000120      .WORD T#CODE
025462 007470      .WORD OPR002
025464 000001      .WORD 1
025466      10000#:
4864 025466 005767 156672      TST    OBUFF ;TEST IF RESPONSE YES
4865 025472 001740      BEQ    2# ;NO - SKIP
4866
4867 025474 004767 171312      1#:   JSR    PC,TSTINT ;INITIALIZE TEST
4868 025500 004767 171324      JSR    PC,GSTATR ;GET STATUS WITH RESET
4869 025504 025556      65#
4870 025506 032767 000040 155336      BIT    #COSTAT,T.MP ;TEST IF COVER OPEN RESET
4871 025514 001406      BEQ    9# ;YES - SKIP
4872 025516 012703 010562      MOV    #MCOSTA,R3 ;SET NAME MESSAGE POINTER
4873 025522      ERRHRD 201...ERR2
025522 104456      TRAP   C#ERRHD
025524 000311      .WORD 201
025526 000000      .WORD 0
025530 012532      .WORD ERR2
4874
4875 025532 032767 020000 155312 9#:   BIT    #MLSTAT,T.MP ;TEST IF WRITE LOCK RESET
4876 025540 001406      BEQ    65# ;YES - SKIP
4877 025542 012703 010610      MOV    #MMLSTA,R3 ;SET NAME MESSAGE POINTER
4878 025546      ERRHRD 202...ERR2
025546 104456      TRAP   C#ERRHD
025550 000312      .WORD 202
025552 000000      .WORD 0
025554 012532      .WORD ERR2
4879 025556      65#:
4880 025556      ENDTST
025556      L10025:
025556 104401      TRAP   C#ETST
4881
4882      .SBTTL *TEST 3 HEAD LOADING
4883 025560      BGNTST ;TEST03
4884
4885      T3::
4886 025560 005767 155574      TST    PASNUM ;SPIN UP THE DRIVE. VERIFY THAT THE DRIVE GOES FROM
4887 025564 001003      BNE    1# ;STATE 0 TO STATE 5 PROPERLY.
4888 025566 005767 166534      TST    MISWIW ;TEST IF PASS 0
4889 025572 100402      BMI    2# ;NO - SKIP
4890 025574      1#:   EXIT   TST ;TEST IF MANUAL INTERVENTION
025574 104432      TRAP   C#EXIT ;YES - SKIP
025576 001500      .WORD L10026-
4891 025600 004767 171206      2#:   JSR    PC,TSTINT ;INITIALIZE TEST
4892 025604 004767 171220      JSR    PC,GSTATR ;GET STATUS
4893 025610 027276      T365#
    
```

*TEST 3

HEAD LOADING

```

4894 025612 005767 155242      TST      T.STAT      ;TEST IF STATE 0
4895 025616 001426              BEQ      4#          ;YES - SKIP
4896 025620              3#:              ;PRINT UNEXPECTED ERROR AND EXIT TEST
4897 025620              PRINTF   #FMTOP1,#UNXERR,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      025620 005046              CLR      -(SP)
      025622 156716 155205      BISB    RLDRV+1,(SP)
      025626 012746 006053      MOV     #DRVNAM,-(SP)
      025632 016746 155170      MOV     RLBAS,-(SP)
      025636 012746 006042      MOV     #BASADD,-(SP)
      025642 012746 010135      MOV     #OPR1A,-(SP)
      025646 012746 006350      MOV     #UNXERR,-(SP)
      025652 012746 011442      MOV     #FMTOP1,-(SP)
      025656 012746 000007      MOV     #7,-(SP)
      025662 010600      MOV     SP,R0
      025664 104417      TRAP   C#PNTF
      025666 062706 000020      ADD    #20,SP
4898 025672 104401      TRAP   C#ETST
4899
4900 025674              4#:              ;PROMPT OPERATOR TO "PRESS LOAD"
4901 025674              PRINTF   #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      025674 005046              CLR      -(SP)
      025676 156716 155131      BISB    RLDRV+1,(SP)
      025702 012746 006053      MOV     #DRVNAM,-(SP)
      025706 016746 155114      MOV     RLBAS,-(SP)
      025712 012746 006042      MOV     #BASADD,-(SP)
      025716 012746 010135      MOV     #OPR1A,-(SP)
      025722 012746 007651      MOV     #OPR3,-(SP)
      025726 012746 011442      MOV     #FMTOP1,-(SP)
      025732 012746 000007      MOV     #7,-(SP)
      025736 010600      MOV     SP,R0
      025740 104417      TRAP   C#PNTF
      025742 062706 000020      ADD    #20,SP
4902
4903 025746 012767 000004 155030      MOV     #CYLUP,OPFLAG ;SET CYCLE UP FLAG
4904 025754 012703 000001      MOV     #1,R3         ;SET EXPECTED STATE VALUE
4905 025760 012767 006416 155024      MOV     #NSTACHG,ERHEAD ;SET ERROR HEADER
4906 025766 012701 000454      MOV     #300.,R1      ;WAIT COUNT R1*TIMDLY= 30 SECONDS.
4907 025772 004767 171046      6#:      JSR     PC,GSTATC     ;GET STATUS
4908 025776 027276              T365#
4909 026000 005767 155054      TST     T.STAT      ;TEST IF STATE IS STILL 0
4910 026004 001034              BNE     10#         ;NO - SKIP
4911 026006 005301              DEC     R1          ;DEC WAIT COUNT
4912 026010 001415              BEQ     7#          ;EXIT IF WAIT DONE
4913
4914 026012              ;              ;JSD REV A
4915 026042 000753              ;              ;JSD REV A
4916
4917 026044 005067 156314      7#:      CLR     OBUFF       ;CLEAR FOR RESPONSE
4918 026050              GMANIL  OPRO03,OBUFF,1,NO
      026050 104443              TRAP   C#GMAN
      026052 000404              BR     10000#
      026054 004364              .WORD OBUFF
      026056 000120              .WORD T#CODE
      026060 007515              .WORD OPRO03
      026062 000001              .WORD 1
4919 026064 005767 156274      10000#: TST     OBUFF       ;TEST IF RESPONSE YES
    
```

*TEST 3

HEAD LOADING

4920	026070	001005			BNE	11:			;YES - REPORT
4921	026072	000167	177522		JMP	3:			
4922	026076	020367	154756	10:	CMP	R3,T,STAT			;CHECK IF NOW STATE 1
4923	026102	001406			BEQ	13:			;YES - SKIP
4924	026104			11:	ERRHRD	301...ERR7			
	026104	104456			TRAP	C#ERHRD			
	026106	000455			.WORD	301			
	026110	000000			.WORD	0			
	026112	013666			.WORD	ERR7			
4925	026114				EXIT	TST			
	026114	104432			TRAP	C#EXIT			
	026116	001160			.WORD	L10026..			
4926	026120	012701	000454	13:	MOV	#300,R1			;INITIALIZE WAIT COUNT FOR 30 SECONDS
4927	026124	012703	000002		MOV	#2,R3			;SET EXPECTED STATE VALUE
4928	026130	004767	170710	14:	JSR	PC,GSTATC			;GET STATUS
4929	026134	027276			T365:				
4930	026136	020367	154716		CMP	R3,T,STAT			;CHECK IF STATE 2
4931	026142	001445			BEQ	20:			;YES - GO TO STATE 2
4932	026144	002002			BGE	17:			;CHECK IF NO CHANGE CONTINUE WAIT
4933	026146	000167	000420		JMP	32:			;GO TO STATE 3.
4934	026152	005301		17:	DEC	R1			;DEC WAIT COUNT
4935	026154	001415			BEQ	18:			;SKIP IF 0
4936				:	TIMDLY	1000.			;JSD REV A
4937	026156				WAITUS	1000.			;JSD REV A
4938	026206	000750			BR	14:			;CHECK FOR STATE CHANGE
4939	026210			18:	ERRHRD	303...ERR7			
	026210	104456			TRAP	C#ERHRD			
	026212	000457			.WORD	303			
	026214	000000			.WORD	0			
	026216	013666			.WORD	ERR7			
4940	026220	032767	004000	154624	BIT	#SPDSTAT,T.MP			;TEST IF SPINDLE TIMEOUT
4941	026226	001011			BNE	19:			;YES - SKIP
4942	026230	012767	006430	154554	MOV	#SPDERR,ERHEAD			;SET ERROR HEADER
4943	026236	012703	010662		MOV	#MSPERR,R3			;SET NAME MESSAGE POINTER
4944	026242				ERRHRD	304...ERR3			
	026242	104456			TRAP	C#ERHRD			
	026244	000460			.WORD	304			
	026246	000000			.WORD	0			
	026250	012600			.WORD	ERR3			
4945	026252			19:	EXIT	TST			
	026252	104432			TRAP	C#EXIT			
	026254	001022			.WORD	L10026..			
4946									
4947	026256	012701	000005	20:	MOV	#5,R1			;WAIT .5 SECONDS
4948				21:	TIMDLY	1000.			;JSD REV A
4949	026262			21:	WAITUS	1000.			;JSD REV A
4950	026312	005301			DEC	R1			
4951	026314	001362			BNE	21:			
4952									
4953	026316	004767	170522		JSR	PC,GSTATC			;CHECK TO SEE IF STATE 3. IF YES GO TO STATE 3
4954	026322	027276			T365:				
4955	026324	022767	000003	154526	CMP	#3,T,STAT			
4956	026332	003002			BGT	22:			
4957	026334	000167	000232		JMP	32:			
4958									
4959	026340	012767	006373	154444	22:	MOV	#MISTST,ERHEAD		;SET ERROR HEADER
4960	026346	012704	011323		MOV	#STATE2,R4			;SET CONDITION MESSAGE POINTER

*TEST 3 HEAD LOADING

4961	026352	012703	010575			MOV	#MBHSTA,R3	;SET NAME MESSAGE POINTER
4962	026356	032767	000010	154466		BIT	#BHSTAT,T.MP	;TEST IF BRUSH HOME STILL SET
4963	026364	001006				BNE	23#	;YES - SKIP
4964	026366					ERRHRD	305...ERR5	
	026366	104456				TRAP	C#ERHRD	
	026370	000461				.WORD	305	
	026372	000000				.WORD	0	
	026374	012716				.WORD	ERR5	
4965	026376					EXIT	TST	
	026376	104432				TRAP	C#EXIT	
	026400	000676				.WORD	L10026--	
4966	026402	012701	000062		23#:	MOV	#50.,R1	;SET WAIT COUNT FOR 5 SECONDS
4967	026406	004767	170432		24#:	JSR	PC,GSTATC	;GET STATUS
4968	026412	027276				T365#		
4969	026414	032767	000010	154430		BIT	#BHSTAT,T.MP	;TEST IF BRUSH HOME RESET
4970	026422	001425				BEQ	27#	;YES - SKIP
4971	026424	005301				DEC	R1	;DEC WAIT COUNT
4972	026426	001415				BEQ	26#	;SKIP IF ZERO
4973						TIMDLY	1000.	;JSD REV A
4974	026430					WAITUS	1000.	;JSD REV A
4975	026460	000752				BR	24#	;LOOP
4976	026462				26#:	ERRHRD	306...ERR4	
	026462	104456				TRAP	C#ERHRD	
	026464	000462				.WORD	306	
	026466	000000				.WORD	0	
	026470	012646				.WORD	ERR4	
4977	026472					EXIT	TST	
	026472	104432				TRAP	C#EXIT	
	026474	000602				.WORD	L10026--	
4978	026476	012701	000454		27#:	MOV	#300.,R1	;INITIALIZE WAIT COUNT FOR 30 SECONDS
4979	026502	004767	170336		28#:	JSR	PC,GSTATC	;GET STATUS
4980	026506	027276				T365#		
4981	026510	032767	000010	154334		BIT	#BHSTAT,T.MP	;TEST IF BRUSH HOME SET AGAIN
4982	026516	001025				BNE	32#	;YES - SKIP
4983	026520	005301				DEC	R1	;ELSE DEC WAIT COUNT
4984	026522	001415				BEQ	30#	;SKIP IF 0
4985						TIMDLY	1000.	;JSD REV A
4986	026524					WAITUS	1000.	;JSD REV A
4987	026554	000752				BR	28#	
4988	026556				30#:	ERRHRD	307...ERR5	
	026556	104456				TRAP	C#ERHRD	
	026560	000463				.WORD	307	
	026562	000000				.WORD	0	
	026564	012716				.WORD	ERR5	
4989	026566					EXIT	TST	
	026566	104432				TRAP	C#EXIT	
	026570	000506				.WORD	L10026--	
4990	026572	012767	006416	154212	32#:	MOV	#NSTACHG,ERHEAD	;SET ERROR HEADER
4991	026600	012703	000003			MOV	#3,R3	;SET EXPECTED STATE VALUE
4992	026604	004767	170234			JSR	PC,GSTATC	;GET STATUS
4993	026610	027276				T365#		
4994	026612	020367	154242			CMP	R3,T.STAT	;CHECK IF STATE 3
4995	026616	001406				BEQ	36#	;YES - SKIP
4996	026620					ERRHRD	308...ERR7	
	026620	104456				TRAP	C#ERHRD	
	026622	000464				.WORD	308	
	026624	000000				.WORD	0	

*TEST 3 HEAD LOADING

4997	026626	013666				.WORD	ERR7		
	026630					EXIT	TST		
	026630	104432				TRAP	C#EXIT		
	026632	000444				.WORD	L10026-		
4998	026634	012767	006373	154150	36#:	MOV	#MISTST,ERHEAD	;SET ERROR HEADER	
4999	026642	012704	011333			MOV	#STATE3,R4	;SET CONDITION MESSAGE POINTER	
5000	026646	012703	010621			MOV	#MHOSTA,R3	;SET NAME MESSAGE POINTER	
5001	026652	004767	170166			JSR	PC,GSTATC	;GET STATUS	
5002	026656	027276				T365#			
5003	026660	032767	000020	154164		BIT	#HOSTAT,T.MP	;TEST IF HEADS OUT SET	
5004	026666	001006				BNE	38#	;YES - SKIP	
5005	026670					ERRHRD	309...ERR5		
	026670	104456				TRAP	C#ERHRD		
	026672	000465				.WORD	309		
	026674	000000				.WORD	0		
	026676	012716				.WORD	ERR5		
5006	026700					EXIT	TST		
	026700	104432				TRAP	C#EXIT		
	026702	000374				.WORD	L10026-		
5007	026704	012701	005670		38#:	MOV	#3000.,R1	;SET WAIT COUNT FOR 300 MS	
5008	026710	012767	006416	154074		MOV	#NSTACHG,ERHEAD	;SET ERROR HEADER	
5009	026716	012703	000004			MOV	#4,R3	;SET EXPECTED STATE VALUE	
5010	026722	004767	170116		43#:	JSR	PC,GSTATC	;GET STATUS	
5011	026726	027276				T365#			
5012	026730	020367	154124			CMP	R3,T.STAT	;CHECK IF STATE 4	
5013	026734	001425				BEQ	49#	;YES - SKIP	
5014	026736	005301				DEC	R1	;DEC WAIT COUNT	
5015	026740	001415				BEQ	47#	;SKIP IF 0	
5016						TIMDLY	1		;JSD REV A
5017	026742					WAITUS	1		;JSD REV A
5018	026772	000753				BR	43#		
5019	026774				47#:	ERRHRD	312...ERR7		
	026774	104456				TRAP	C#ERHRD		
	026776	000470				.WORD	312		
	027000	000000				.WORD	0		
	027002	013666				.WORD	ERR7		
5020	027004					EXIT	TST		
	027004	104432				TRAP	C#EXIT		
	027006	000270				.WORD	L10026-		
5021	027010	012701	000454		49#:	MOV	#300.,R1	;SET WAIT COUNT FOR 30 MS	
5022	027014	012703	000005			MOV	#5,R3	;SET EXPECTED STATE VALUE	
5023	027020	004767	170020		50#:	JSR	PC,GSTATC	;GET STATUS	
5024	027024	027276				T365#			
5025	027026	020367	154026			CMP	R3,T.STAT	;CHECK IF STATE 5	
5026	027032	001425				BEQ	53#	;YES - SKIP	
5027	027034	005301				DEC	R1	;DEC WAIT COUNT	
5028	027036	001415				BEQ	52#	;ELSE SKIP	
5029						TIMDLY	1		;JSD REV A
5030	027040					WAITUS	1		;JSD REV A
5031	027070	000753				BR	50#		
5032	027072				52#:	ERRHRD	313...ERR7		
	027072	104456				TRAP	C#ERHRD		
	027074	000471				.WORD	313		
	027076	000000				.WORD	0		
	027100	013666				.WORD	ERR7		
5033	027102					EXIT	TST		
	027102	104432				TRAP	C#EXIT		

*TEST 3 HEAD LOADING

```

5034 027104 000172          .WORD L10026-.
5035 027106 032767 001000 153736 53#: BIT #VCSTAT,T.MP ;VOLUME CHECK SHOULD BE SET FOR
5036 027114 001010          BNE 54# ;STATE 5, IF NOT GIVE ERROR.
5037 027116 012703 010551 MOV #MVOLCK,R3 ;SET NAME MESSAGE POINTER
027122 104456 ERRHRD 310.,,ERR5
027124 000466 TRAP C#ERHRD
027126 000000 .WORD 310
027130 012716 .WORD 0
5038 027132 EXIT ERR5
027132 104432 TRAP TST
027134 000142 TRAP C#EXIT
5039 027136 032767 040000 153700 54#: .WORD L10026-.
5040 027144 001010          BIT #DRVERR,T.CS ;TEST IF DRIVE ERROR SET
5041 027146 012703 010526 BNE 57# ;YES - SKIP
5042 027152 MOV #MDRERR,R3 ;SET NAME MESSAGE POINTER
027152 104456 ERRHRD 315.,,ERR5
027154 000473 TRAP C#ERHRD
027156 000000 .WORD 315
027160 012716 .WORD 0
5043 027162 EXIT ERR5
027162 104432 TRAP TST
027164 000112 TRAP C#EXIT
5044 027166 012701 000120          .WORD L10026-.
5045 027172 004767 167646          MOV #80.,R1 ;SET WAIT FOR 8 MS
5046 027176 027276          JSR PC,GSTATC ;GET STATUS
5047 027200 032767 000001 153636 T365#
5048 027206 001033          BIT #DRDYMSK,T.CS ;CHECK IF DRIVE READY
5049 027210 005301          BNE 172# ;YES - SKIP
5050 027212 001415          DEC R1 ;DEC COUNT
5051 TIMDLY 1 ;SKIP IF 0
5052 WAITUS 1 ;JSD REV A
5053 BR 56# ;JSD REV A
5054 027244 000752          BR 56#
5055 027246 012767 006373 153536 58#: MOV #MISTST,ERHEAD ;SET ERROR HEADER
5056 027254 012704 011343 MOV #STAT5,R4 ;SET CONDITION MESSAGE POINTER
5057 027260 012703 010404 MOV #MDRDY,R3 ;SET NAME MESSAGE POINTER
027264 104456 ERRHRD 316.,,ERR5
027266 000474 TRAP C#ERHRD
027270 000000 .WORD 316
027272 012716 .WORD 0
5058 027274 000400          .WORD ERR5
5059 027276          BR 172# ;EXIT TEST
5060 027276          T365#:
5061 027276          ENDTST
027276 104401          L10026: TRAP C#ETST
5062
5063          .SBTTL *TEST 4 HEAD UNLOADING
5064 027300          BGNTST ;TEST04
5065          T4::
5066          ;SPIN DOWN AND UNLOAD HEADS. VERIFY THAT THE DRIVE
5067 027300 005767 154054          TST PASNUM ;TEST IF FIRST PASS
5068 027304 001003          BNE 8# ;NO - SKIP
5069 027306 005767 165014          TST MISWIW ;TEST IF MANUAL INTERVENTION
5070 027312 100403          BMI TST4 ;YES - SKIP

```

*TEST 4 HEAD UNLOADING

```

5071 027314          8#:  EXIT  TST
      027314 104432      TRAP  C#EXIT
      027316 000662      .WORD L10027-.

5072
5073 027320          BGNSUB
      027320
      027320 104402
5074 027322 012767 006416 153462 TST4: TRAP  C#BSUB
5075 027330 004767 167456      MOV  #NSTACHG,ERHEAD ;SET ERROR HEADER
5076 027334 004767 167470      JSR  PC,TSTINT      ;INITIALIZE TEST
5077 027340 030070      JSR  PC,GSTATR     ;GET STATUS
5078 027342 032767 000001 153474      BIT  #DRDYMSK,T.CS ;CHECK IF DRIVE READY
5079 027350 001040      BNE  3#           ;YES - SKIP
5080
5081 027352          1#:  PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      027352 005046      CLR  -(SP)
      027354 156716 153453      BISB RLDRV+1,(SP)
      027360 012746 006053      MOV  #DRVNAM,-(SP)
      027364 016746 153436      MOV  RLBAS,-(SP)
      027370 012746 006042      MOV  #BASADD,-(SP)
      027374 012746 010135      MOV  #OPR1A,-(SP)
      027400 012746 007665      MOV  #OPR6,-(SP)
      027404 012746 011442      MOV  #FMTOP1,-(SP)
      027410 012746 000007      MOV  #7,-(SP)
      027414 010600      MOV  SP,RO
      027416 104417      TRAP C#PNTF
      027420 062706 000020      ADD  #20,SP
5082 027424 005067 154734      CLR  OBUFF ;CLEAR FOR RESPONSE
5083 027430          GMANIL OPR002,OBUFF,1,NO
      027430 104443      TRAP C#GMAN
      027432 000404      BR   10000#
      027434 004364      .WORD OBUFF
      027436 000120      .WORD T#CODE
      027440 007470      .WORD OPR002
      027442 000001      .WORD 1
      027444          10000#:
5084 027444 005767 154714      TST  OBUFF ;TST RESPONSE YES
5085 027450 001740      BEQ  1#       ;NO - SKIP
5086
5087 027452 052767 000010 153324 3#:  BIS  #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
5088
5089 027460          4#:  PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      027460 005046      CLR  -(SP)
      027462 156716 153345      BISB RLDRV+1,(SP)
      027466 012746 006053      MOV  #DRVNAM,-(SP)
      027472 016746 153330      MOV  RLBAS,-(SP)
      027476 012746 006042      MOV  #BASADD,-(SP)
      027502 012746 010135      MOV  #OPR1A,-(SP)
      027506 012746 007651      MOV  #OPR3,-(SP)
      027512 012746 011442      MOV  #FMTOP1,-(SP)
      027516 012746 000007      MOV  #7,-(SP)
      027522 010600      MOV  SP,RO
      027524 104417      TRAP C#PNTF
      027526 062706 000020      ADD  #20,SP
5090
5091 027532 012703 000006      MOV  #6,R3 ;SET EXPECTED STATE VALUE
5092 027536 012704 000144      MOV  #100.,R4 ;SET SECOND LEVEL COUNT

```

T4.1:

```

*TEST 4      HEAD UNLOADING

5093 027542 012701 001274      MOV      #700.,R1      ;SET WAIT COUNT FOR 30 SECONDS
5094 027546 004767 167272      JSR      PC,GSTATC    ;GET STATUS
5095 027552 030070      T465#
5096 027554 020367 153300      CMP      R3,T.STAT    ;WATCH FOR STATE CHANGE FROM 5 TO 6
5097 027560 001450      BEQ      11#          ;YES - SKIP
5098 027562 022767 000005 153270  CMP      #5,T.STAT    ;TEST IF STATE 5
5099 027570 001036      BNE      9#          ;NO - REPORT WRONG STATE
5100 027572 005304      8#:     DEC      R4      ;DEC 2ND LEVEL COUNT
5101 027574 001004      BNE      6#          ;SKIP IF NOT 0
5102 027576 005301      DEC      R1          ;ELSE DEC 1ST LEVEL COUNT
5103 027600 001417      BEQ      7#          ;IF 0 - SKIP TO QUESTION
5104 027602 012704 000144      MOV      #100.,R4    ;ELSE RESET 2ND LEVEL
5105      ;6#:     TIMDLY    1      ;WAIT 100 US      ;JSD REV A
5106      6#:     WAITUS   1      ;WAIT 100 US      ;JSD REV A
5107 027636 000743      BR       5#
5108 027640 005067 154520      7#:     CLR      OBUFF    ;CLEAR FOR RESPONSE
5109 027644      GMANIL   OPR003,OBUFF,1,NO
      027644 104443      TRAP     C#GMAN
      027646 000404      BR       10001#
      027650 004364      .WORD   OBUFF
      027652 000120      .WORD   T#CODE
      027654 007515      .WORD   OPR003
      027656 000001      .WORD   1
5110 027660 005767 154500      10001#: TST      OBUFF        ;TEST IF RESPONSE YES
5111 027664 001675      BEQ      4#          ;NO - SKIP
5112 027666      9#:     ERRHRD   401,,,ERR7 ;ELSE REPORT STATE CHANGE WRONG
      027666 104456      TRAP     C#ERRRD
      027670 000621      .WORD   401
      027672 000000      .WORD   0
      027674 013666      .WORD   ERR7
5113 027676      EXIT     SUB
      027676 104432      TRAP     C#EXIT
      027700 000176      .WORD   L10030-.
5114 027702 012703 000007      11#:    MOV      #7,R3      ;SET EXPECTED STATE VALUE
5115 027706 012701 005670      MOV      #3000.,R1   ;SET COUNT FOR 300MS
5116 027712 004767 167126      12#:    JSR      PC,GSTATC ;GET STATUS
5117 027716 030070      T465#
5118 027720 020367 153134      CMP      R3,T.STAT    ;CHECK IF STATE 7
5119 027724 001425      BEQ      18#        ;YES - SKIP
5120 027726 005301      DEC      R1          ;DEC WAIT COUNT
5121 027730 001415      BEQ      16#        ;TIME OUT GIVE ERROR MESSAGE
5122      ;      TIMDLY    1      ;JSD REV A
5123 027732      WAITUS   1      ;JSD REV A
5124 027762 000753      BR       12#
5125 027764      16#:    ERRHRD   402,,,ERR7 ;REPORT WRONG STATE CHANGE
      027764 104456      TRAP     C#ERRRD
      027766 000622      .WORD   402
      027770 000000      .WORD   0
      027772 013666      .WORD   ERR7
5126 027774      EXIT     SUB
      027774 104432      TRAP     C#EXIT
      027776 000100      .WORD   L10030-.
5127 030000 005003      18#:    CLR      R3          ;SET EXPECTED STATE VALUE
5128 030002 012701 013560      MOV      #6000.,R1   ;SET WAIT COUNT FOR 60 SECONDS
5129 030006 004767 167032      20#:    JSR      PC,GSTATC ;GET STATUS
5130 030012 030070      T465#
    
```

*TEST 4 HEAD UNLOADING

```

5131 030014 005767 153040      TST      T,STAT      ;CHECK IF STATE 0
5132 030020 001423              BEQ      24$         ;YES - SKIP
5133 030022 005301              DEC      R1         ;DEC WAIT COUNT
5134 030024 001415              BEQ      22$         ;SKIP IF 0
5135                      ;
5136 030026              TIMDLY  100.        ;JSD REV A
5137 030056 000753              WAITUS 100.        ;JSD REV A
5138 030060 22$: ERRHRD 403...ERR7 ;REPORT WRONG STATE CHANGE
      030060 104456      TRAP      C$ERHRD
      030062 000623      .WORD    403
      030064 000000      .WORD    0
      030066 013666      .WORD    ERR7
5139 030070              24$:
5140 030070 012767 000002 152720 T465$: MOV      #2,ERRSWI ;INIT ERROR SWITCH
5141
5142 030076              ENDSUB
      030076              L10030:
      030076 104403      TRAP      C$ESUB
5143                      ;PROMPT PRESS LD AND WAIT FOR RDY
5144 030100 26$: PRINTF #FMTOP1,#OPR6,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
      030100 005046      CLR      -(SP)
      030102 156716 152725 BISH    RLDRV+1,(SP)
      030106 012746 006053 MOV      #DRVNAM,-(SP)
      030112 016746 152710 MOV      RLBAS,-(SP)
      030116 012746 006042 MOV      #BASADD,-(SP)
      030122 012746 010135 MOV      #OPR1A,-(SP)
      030126 012746 007665 MOV      #OPR6,-(SP)
      030132 012746 011442 MOV      #FMTOP1,-(SP)
      030136 012746 000007 MOV      #7,-(SP)
      030142 010600      MOV      SP,R0
      030144 104417      TRAP      C$PNTF
      030146 062706 000020 ADD      #20,SP
5145
5146 030152 005067 154206      CLR      OBUFF      ;CLEAR FOR RESPONSE
5147 030156              GMANIL  OPR002,OBUFF,1,NO
      030156 104443      TRAP      C$GMAN
      030160 000404      BR       10000$
      030162 004364      .WORD    OBUFF
      030164 000120      .WORD    T$CODE
      030166 007470      .WORD    OPR002
      030170 000001      .WORD    1
      10000$:
5148 030172 005767 154166      TST      OBUFF      ;TEST IF RESPONSE YES
5149 030176 001740              BEQ      26$         ;NO - SKIP
5150 030200 29$:
5151
5152 030200              ENDTST
      030200              L10027:
      030200 104401      TRAP      C$ETST
5153
5154                      .SBTTL *TEST 5      DRIVE SELECT
5155 030202              BGNTST ;TEST05
5156 030202 012767 000002 152606 MOV      #2,ERRSWI ;SET FOR NO ERROR RETURN
5157 030210 005767 153144 TST      PASNUM     ;TEST IF FIRST PASS
5158 030214 001173              BNE     EXT05      ;NO - SKIP
5159 030216 032767 000004 164102 BIT      #DRSELT,MISWIW ;TEST IF SELECT TESTS

```

*TEST 5 DRIVE SELECT

```

5160 030224 001567          BEQ     EXT05          ;NO - SKIP
5161 030226          1#: PRINTF  #FMTOP1,#OPR7,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
    030226 005046          CLR     -(SP)
    030230 156716 152577    BISB   RLDRV+1,(SP)
    030234 012746 006053    MOV    #DRVNAM,-(SP)
    030240 016746 152562    MOV    RLBAS,-(SP)
    030244 012746 006042    MOV    #BASADD,-(SP)
    030250 012746 010135    MOV    #OPR1A,-(SP)
    030254 012746 007720    MOV    #OPR7,-(SP)
    030260 012746 011442    MOV    #FMTOP1,-(SP)
    030264 012746 000007    MOV    #7,-(SP)
    030270 010600          MOV    SP,R0
    030272 104417          TRAP   C#PNTF
    030274 062706 000020    ADD    #20,SP

5162 030300 005067 154060          CLR     OBUFF          ;REQUEST "REMOVE ADD PLGS EXCPT "
5163 030304          GMANIL OPR002,OBUFF,1,NO ;CLEAR FOR RESPONSE
5164 030304 104443          TRAP   C#GMAN
    030306 000404          BR     10000#
    030310 004364          .WORD OBUFF
    030312 000120          .WORD T#CODE
    030314 007470          .WORD OPR002
    030316 000001          .WORD 1
    030320          10000#: TST     OBUFF          ;TEST RESPONSE YES
5165 030320 005767 154040          BEQ     1#             ;NO - SKIP
5166 030324 001740          152456 3#: MOV     #TOSERR,ERHEAD ;SET ERROR HEADER MESSAGE
5167 030326 012767 006530          JSR    PC,TSTINT      ;INITIALIZE TEST
5168 030334 004767 166452          JSR    PC,GSTATC     ;DO SELECT AND GET STATUS
5169 030340 004767 166500          T504#
5170 030344 030526          MOV    RLDRV,TEMPO   ;STORE ORIGINAL DRIVE NUMBER
5171 030346 016767 152460          MOV    RLDRV,R1      ;PUT IT IN R1
5172 030354 016701 152452          MOV    #4,R4         ;SET COUNT FOR NUMBER OF PLUGS
5173 030360 012704 000004          LPT05: ADD    #400,R1  ;BUMP TO NEXT DRIVE
5174 030364 062701 000400          CMP    #2000,R1     ;CHECK IF TOO LARGE
5175 030370 022701 002000          BNE    4#            ;NO - SKIP
5176 030374 001001          CLR    R1            ;ELSE CLEAR TO DRIVE 0
5177 030376 005001          MOV    R1,RLDRV     ;PUT IT BACK IN RLDRV
5178 030400 010167 152426          4#: PRINTF #FMTOP3,#OPR8,<B,RLDRV+1>,#OPR1B,#UNDTST
5179 030404          5#: MOV    #UNDTST,-(SP)
    030404 012746 010151    MOV    #OPR1B,-(SP)
    030410 012746 010141    CLR    -(SP)
    030414 005046          BISB   RLDRV+1,(SP)
    030416 156716 152411    MOV    #OPR8,-(SP)
    030422 012746 007747    MOV    #FMTOP3,-(SP)
    030426 012746 011513    MOV    #5,-(SP)
    030432 012746 000005    MOV    SP,R0
    030436 010600          TRAP   C#PNTF
    030440 104417          ADD    #14,SP

5180 030442 062706 000014          ;INSERT PLUG REQUEST
5181 030446 005067 153712          CLR     OBUFF          ;CLEAR FOR RESPONSE
5182 030452          GMANIL OPR002,OBUFF,1,NO
    030452 104443          TRAP   C#GMAN
    030454 000404          BR     10001#
    030456 004364          .WORD OBUFF
    030460 000120          .WORD T#CODE
    030462 007470          .WORD OPR002
    
```

```

*TEST 5          DRIVE SELECT
      030464 000001
      030466
5183 030466 005767 153672
5184 030472 001744
5185 030474
      030474
      030474 104402
5186 030476 004767 166342
5187 030502 030504
5188 030504 012767 000002 152304 60#:
5189
5190 030512
      030512
      030512 104403
5191 030514 005304
5192 030516 001322
5193 030520 016767 152372 152304
5194 030526
5195 030526
      030526 012746 007766
      030532 012746 007747
      030536 012746 011556
      030542 012746 000003
      030546 010600
      030550 104417
      030552 062706 000010
5196 030556 005067 153602
5197 030562
      030562 104443
      030564 000404
      030566 004364
      030570 000120
      030572 007470
      030574 000001
      030576
5198 030576 005767 153562
5199 030602 001751
5200 030604
5201 030604
      030604
      030604 104401
5202
5203
5204 030606
      030606
5205 030606 005767 152546
5206 030612 001004
5207 030614 032767 000004 163504
5208 030622 001002
5209 030624
      030624 104432
      030626 000744
5210 030630 012767 006464 152154 6#:
5211 030636 004767 166150
5212 030642 016703 152514
5213 030646 026727 151140 000001
5214 030654 101461

      10001#: .WORD 1
      TST OBUFF ;TEST RESPONSE YES
      BEQ 5# ;NO - SKIP
      BGNSUB
      TRAP C#BSUB T5.1:
      JSR PC,GSTATC ;GET STATUS - REPORT ANY ERROR
      60# MOV #2,ERRSWI ;INIT ERROR SWITCH
      ENDSUB
      L10032:
      TRAP C#ESUB
      DEC R4 ;DEC COUNT
      BNE LPT05 ;LOOP IF NOT ZERO
      MOV TEMPO,RLDRV ;ELSE RESTORE RLDRV
      T504#:
      4#: PRINTF #FMT4,#OPR8,#OPR9
      MOV #OPR9,-(SP)
      MOV #OPR8,-(SP)
      MOV #FMT4,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C#PNTF
      ADD #10,SP
      CLR OBUFF ;CLEAR FOR RESPONSE
      GMANIL OPR002,OBUFF,1,NO
      TRAP C#GMAN
      BR 10000#
      .WORD OBUFF
      .WORD T#CODE
      .WORD OPR002
      .WORD 1
      10000#:
      TST OBUFF ;TEST IF RESPONSE YES
      BEQ 4# ;NO - SKIP
      EXT05:
      ENDTST
      L10031:
      TRAP C#ETST
      .SBTTL *TEST 6 DRIVE SELECT ERROR TEST
      BGNTST ;TEST06
      TST PASNUM T6::
      BNE 1# ;CHECK IF FIRST PASS
      BIT #DRSELT,MISWIW ;CHECK IF TEST DRIVE SELECT
      BNE 6# ;YES - SKIP
      1#: EXIT TST
      TRAP C#EXIT
      .WORD L10033-
      6#: MOV #GSTER1,ERHEAD ;SET ERROR HEADER
      JSR PC,TSTINT ;INITIALIZE TEST
      MOV PSETNM,R3 ;GET PARAM SET NUMBER
      CMP L#UNIT,#1 ;TEST IF MORE THAN 1 UNIT
      BLOS 5# ;NO - SKIP
    
```



```

*TEST 6      DRIVE SELECT ERROR TEST

5215 030656 005203          2#:  INC      R3          ;BUMP PARAMETER SET NUMBER
5216 030660 020367 151126   CMP      R3,L#UNIT    ;CHECK IF PAST VALID PARAMETER TABLE
5217 030664 101401          BLOS    3#           ;NO - SKIP
5218 030666 005003          CLR      R3          ;ELSE CLEAR TO POINT TO ENTRY 0
5219 030670          3#:  GPHARD  R3,R0
      030670 010300      MOV      R3,R0
      030672 104442      TRAP    C#GPHRD
5220 030674          BNCMPLE 2#          ;SKIP IF NOT AVAILABLE
      030674 103370      BCC     2#
5221 030676 010004          MOV      R0,R4          ;PUT POINTER INTO R4
5222 030700 021467 152122   CMP      (R4),RLBAS   ;CHECK IF SAME CONTROLLER
5223 030704 001364          BNE     2#           ;NO - SKIP
5224 030706 005067 152074   CLR      DONE        ;CLEAR DONE FLAG
5225 030712 012767 000104 152114  MOV      #GTSTAT,L.CS ;LOAD GET STATUS
5226 030720 056467 000010 152106  BIS      10(R4),L.CS  ;INSERT DRIVE
5227 030726 012767 000013 152104  MOV      #GETSTAT:DRSET,L.DA ;SET UP TO CLEAR DRIVE
5228 030734 016762 152100 000004  MOV      L.DA,RLDA(R2) ;LOAD DA REG
5229 030742 016762 152066 000000  MOV      L.CS,RLCS(R2) ;LOAD CS REG
5230          TIMDLY 30.          ;WAIT 3 MS ;JSD REV A
5231 030750          WAITUS 30.          ;WAIT 3 MS ;JSD REV A
5232 031000 005767 152002   TST     DONE        ;TEST IF INTERRUPT
5233 031004 001724          BEQ     2#           ;NO - SKIP
5234 031006 032767 100000 152030  BIT      #ANYERR,T.CS ;TEST IF ANY ERROR SET
5235 031014 001415          BEQ     7#           ;NO - GO TEST
5236 031016 000717          BR     2#           ;ELSE CHECK NEXT DRIVE
5237 031020          5#:  PRINTF  #FMT9,#OPR10 ;REPORT CAN'T FIND 2ND DRIVE
      031020 012746 010003      MOV      #OPR10,-(SP)
      031024 012746 011753      MOV      #FMT9,-(SP)
      031030 012746 000002      MOV      #2,-(SP)
      031034 010600          MOV      SP,R0
      031036 104417          TRAP    C#PNTF
      031040 062706 000006      ADD     #6,SP
5238 031044 000167 000522          JMP     LCLEXT
5239 031050 016467 000010 152042  7#:  MOV      10(R4),TEMP1 ;STORE NEW ADDRESS
5240          ;ASK FOR PLUG CHANGE
5241 031056 016700 151750          9#:  MOV      RLDRV,R0    ;GET DRIVE UNDER TEST
5242 031062 016705 152032          MOV      TEMP1,R5    ;GET NEW ADDRESS
5243 031066 042700 002000          BIC     #2000,R0    ;CLEAR FOR ADDRESS 0 TO 3
5244 031072 042705 002000          BIC     #2000,R5
5245 031076 020527 001400          20#:  CMP     R5,#1400   ;TEST IF DRIVE NUMBER 3
5246 031102 001001          BNE     21#        ;NO - SKIP
5247 031104 005005          CLR     R5         ;ELSE SET TO DRIVE NUMBER 0
5248 031106 062705 000400          21#:  ADD     #400,R5   ;BUMP TO NEXT ADDRESS
5249 031112 020500          CMP     R5,R0     ;THIS EQUAL TO NEW ADDRESS?
5250 031114 001770          BEQ     20#       ;YES - SKIP
5251 031116 052705 000200          BIS     #CRDYMSK,R5 ;ELSE SET CONTROLLER READY BIT
5252 031122 010562 000000          MOV     R5,RLCS(R2) ;AND LOAD CS REG
5253          ;PROMPT INSRT ADR PLG AN DRV
5254 031126          PRINTF #FMTOP2,#OPR8,<B,RLDRV+1>,#OPR1B,<B,TEMP1+1>
      031126 005046          CLR     -(SP)
      031130 156716 151765          BISB   TEMP1+1,(SP)
      031134 012746 010141          MOV     #OPR1B,-(SP)
      031140 005046          CLR     -(SP)
      031142 156716 151665          BISB   RLDRV+1,(SP)
      031146 012746 007747          MOV     #OPR8,-(SP)
      031152 012746 011471          MOV     #FMTOP2,-(SP)
      031156 012746 000005          MOV     #5,-(SP)

```

*TEST 6 DRIVE SELECT ERROR TEST

```

031162 010600      MOV      SP,R0
031164 104417      TRAP     C:PNTF
5255 031166 062706 000014  ADD     @14,SP
5256 031172 005067 153166  CLR     OBUFF ;CLEAR FOR RESPONSE
031176 104443      GMANIL  OPRO02,OBUFF,1,NO
031200 000404      TRAP     C:GMAN
031202 004364      BR      10000$
031204 000120      .WORD   OBUFF
031206 007470      .WORD   T:CODE
031210 000001      .WORD   OPRO02
031212 10000$:      .WORD   1
5257 031212 005767 153146  TST     OBUFF ;TEST IF RESPONSE YES
5258 031216 001717      BEQ     9$ ;NO - SKIP
5259 031220 012704 000012  MOV     @10.,R4 ;SET COUNT
5260 031224      BGNSUB
031224 104402      TRAP     C:BSUB ;T6.1:
5261 031226 016767 151600 151600 8$:      MOV     RLDRV,L.CS ;SET UP TO SELECT MULTIPLE DRIVES
5262 031234 016762 151574 000000  MOV     L.CS,RLCSR(R2) ;DO IT
5263 031242      TIMDLY  100. ;JSD REV A
5264 031272 052767 000104 151534  WAITUS  100. ;JSD REV A
5265 031300 012767 000013 151532  BIS     @GTSTAT,L.CS ;SET GET STATUS
5266 031300 012767 000013 151532  MOV     @GETSTAT!DRSET,L.DA ;SET RESET BIT 3 IN THE DA REG FOR THE
5267 ;/DRIVE TO CLEAR ITS ERROR REGISTER
5268 ;/BEFORE SENDING A STATUS WORD TO THE
5269 ;/MP REG DURING GET STATUS COMMAND
5270
5271 031306 016762 151526 000004  MOV     L.DA,RLDA(R2)
5272 031314 005067 151466      CLR     DONE
5273 031320 016762 151510 000000  MOV     L.CS,RLCSR(R2) ;DO GET STATUS
5274 031326      WAITUS  1 ;WAIT FOR INTERRUPT
5275 031356 005767 151424      TST     DONE ;CHECK IF INTERRUPTED
5276 031362 001012      BNE    12$ ;YES - SKIP
5277 031364 004767 165246      JSR     PC,WAITIN ;WAIT FOR TIMEOUT
5278 031370 012603      MOV     (SP),R3 ;GET ERROR POINTER
5279 031372 001406      BEQ     12$ ;SKIP IF 0
5280 031374      ERRHRD  601.,GSTER1,ERR1
031374 104456      TRAP     C:ERHRD
031376 001131      .WORD   601
031400 006464      .WORD   GSTER1
031402 012464      .WORD   ERR1
5281 031404      EXIT    SUB
031404 104432      TRAP     C:EXIT
031406 000110      .WORD   L10034-.
5282 ;12$:      TIMDLY  20. ;WAIT FOR DSE TO SET ;JSD REV A
5283 ;12$:      WAITUS  20. ;WAIT FOR DSE TO SET ;JSD REV A
5284 031410      JSR     PC,GDRSTA
5285 031444 004767 166462 151400  BIT     @DSESTAT,T.MP ;GET STATUS
5286 031452 001010      BNE    16$ ;TEST IF DRIVE SELECT ERROR SET
5287 031454 012703 010632      MOV     @MDSERR,R3 ;YES - SKIP
5288 031460      ERRHRD  602.,ERR3 ;SET NAME MESSAGE POINTER
031460 104456      TRAP     C:ERHRD
031462 001132      .WORD   602
031464 000000      .WORD   0
031466 012600      .WORD   ERR3
5289 031470      EXIT    SUB
    
```

*TEST 6 DRIVE SELECT ERROR TEST

```

031470 104432 TRAP C#EXIT
031472 000024 .WORD L10034-.
5290 031474 010562 000000 16#: MOV R5,RLC5(R2) ;LOAD IN DIFFERENT ADDRESS
5291 031500 005304 DEC R4 ;DEC COUNT
5292 031502 001402 BEQ 60# ;LOOP IF NOT ZERO
5293 031504 000167 177516 JMP 8#
5294 031510 012767 000002 151300 60#: MOV #2,ERRSWI ;INIT ERROR SWITCH
5295 031516 ENDSUB
031516 L10034:
031516 104403 TRAP C#ESUB
5296 031520 15#: PRINTF #FMT9,#OPR11 ;REQUEST PLUG CHANGE
031520 012746 010051 MOV #OPR11,-(SP)
031524 012746 011753 MOV #FMT9,-(SP)
031530 012746 000002 MOV #2,-(SP)
031534 010600 MOV SP,R0
031536 104417 TRAP C#PNTF
5297 031540 062706 000006 ADD #6,SP
5298 031544 005067 152614 CLR OBUFF ;CLEAR FOR RESPONSE
031550 GMANIL OPRO02,OBUFF,1,NO
031550 104443 TRAP C#GMAN
031552 000404 BR 10000#
031554 004364 .WORD OBUFF
031556 000120 .WORD T#CODE
031560 007470 .WORD OPRO02
031562 000001 .WORD 1
031564 10000#:
5299 031564 005767 152574 TST OBUFF ;TEST IF RESPONSE YES
5300 031570 001753 BEQ 15# ;NO - SKIP
5301 031572 LCLEXT:
5302 031572 ENDTST
031572 L10033:
031572 104401 TRAP C#ETST
5303 .SBTTL *TEST 7 INITIAL STATE
5304 BGNTST ;TEST 07
5305 031574 T7::
031574 TST PASNUM ;CHECK IF FIRST PASS
5306 031574 005767 151560 BNE 1# ;NO - EXIT TEST
5307 031600 001003 TST MISWIW ;CHECK IF MANUAL INTERVENTION
5308 031602 005767 162520 BMI 3# ;PERFORM TEST IF MANUAL INTERVENTION
5309 031606 100402
5310 031610 1#: EXIT TST
031610 104432 TRAP C#EXIT
031612 000556 .WORD L10035-.
5311 031614 012767 006515 151170 3#: MOV #INITST,ERHEAD ;SET ERROR HEADER
5312 031622 004767 165164 JSR PC,TSTINT ;INITIALIZE TEST
5313 ; TIMDLY 10. ;WAIT 1 MS ;JSD REV A
5314 031626 WAITUS 10. ;WAIT 1 MS ;JSD REV A
5315 031656 004767 165146 JSR PC,GSTATR ;GET STATUS WITH RESET
5316 031662 032370 100#
5317 031664 032767 000001 151152 BIT #DRDYMSK,T.CS ;CHECK IF DRIVE IS READY
5318 031672 001432 BEQ 20# ;BRANCH IF DRIVE IS NOT READY
5319
5320 031674 052767 000010 151102 BIS #UNLOAD,OPFLAG ;SET UNLOAD OPERATION
5321 ;PROMPT OPERATOR TO "PRESS LOAD"
5322 031702 PRINTF #FMTOP1,#OPR3,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
031702 005046 CLR -(SP)
031704 156716 151123 BISB RLDRV+1,(SP)

```

*TEST 7

INITIAL STATE

031710	012746	006053		MOV	#DRVNAM, -(SP)	
031714	016746	151106		MOV	RLBAS, -(SP)	
031720	012746	006042		MOV	#BASADD, -(SP)	
031724	012746	010135		MOV	#OPR1A, -(SP)	
031730	012746	007651		MOV	#OPR3, -(SP)	
031734	012746	011442		MOV	#FMTOP1, -(SP)	
031740	012746	000007		MOV	#7, -(SP)	
031744	010600			MOV	SP, R0	
031746	104417			TRAP	C#PNTF	
031750	062706	000020		ADD	#20, SP	
5323	031754	012703	000000	MOV	#0, R3	;SET "LOAD CARTRIDGE" STATE VALUE 0
5324						
5325	031760	004767	165060	20#:	JSR	PC, GSTATC ;GET STATUS
5326	031764	032370			100#	
5327	031766				BREAK	;MAKE A SUPERVISOR CALL
	031766	104422			TRAP	C#BRK
5328	031770	022767	000000	151062	TRAP	C#BRK
5329	031776	001370			TRAP	C#BRK
5330					TRAP	C#BRK
5331					TRAP	C#BRK
5332					TRAP	C#BRK
5333	032000			21#:	PRINTF	#FMTOP1, #OPR6, #OPR1A, #BASADD, RLBAS, #DRVNAM, <B, RLDRV+1>
	032000	005046			CLR	-(SP)
	032002	156716	151025		BISB	RLDRV+1, (SP)
	032006	012746	006053		MOV	#DRVNAM, -(SP)
	032012	016746	151010		MOV	RLBAS, -(SP)
	032016	012746	006042		MOV	#BASADD, -(SP)
	032022	012746	010135		MOV	#OPR1A, -(SP)
	032026	012746	007665		MOV	#OPR6, -(SP)
	032032	012746	011442		MOV	#FMTOP1, -(SP)
	032036	012746	000007		MOV	#7, -(SP)
	032042	010600			MOV	SP, R0
	032044	104417			TRAP	C#PNTF
	032046	062706	000020		ADD	#20, SP
5334	032052	005067	152306		CLR	OBUFF ;CLEAR FOR RESPONSE
5335	032056				GMANIL	OPR002, OBUFF, 1, NO ;PROMPT OPERATOR FOR RESPONSE
	032056	104443			TRAP	C#GMAN
	032060	000404			BR	10000#
	032062	004364			.WORD	OBUFF
	032064	000120			.WORD	T#CODE
	032066	007470			.WORD	OPR002
	032070	000001			.WORD	1
	032072			10000#:		
5336	032072	005767	152266		TST	OBUFF ;TEST IF RESPONSE IS YES
5337	032076	001740			BEQ	21# ;BRANCH IF NOT READY
5338						
5339	032100	004767	164740	22#:	JSR	PC, GSTATC ;GET STATUS
5340	032104	032370			100#	
5341	032106				BREAK	;MAKE A SUPERVISOR CALL
	032106	104422			TRAP	C#BRK
5342	032110	022767	000005	150742	TRAP	C#BRK
5343	032116	001370			TRAP	C#BRK
5344					TRAP	C#BRK
5345	032120	016701	150726		MOV	T, MP, R1 ;GET MP REG
5346	032124	032701	000020		BIT	#HOSTAT, R1 ;CHECK HEADS OUT
5347	032130	001003			BNE	7# ;YES-SKIP
5348	032132	012703	010621		MOV	#HOSTA, R3 ;SET NAME MESSAGE PTR

```

*TEST 7      INITIAL STATE
5349 032136 000405
5350 032140 032701 000010      7#:
5351 032144 001010
5352 032146 012703 010575      9#:
5353 032152
      032152 104456
      032154 001276
      032156 000000
      032160 012600
5354 032162
      032162 104432
      032164 000204
5355 032166 005767 162134      10#:
5356 032172 100035
5357 032174 005767 151160
5358 032200 001032
5359 032202 032701 000100
5360 032206 001412
5361 032210 012703 010537
5362 032214 012704 011412
5363 032220
      032220 104456
      032222 001277
      032224 000000
      032226 012646
5364 032230
      032230 104432
      032232 000136
5365 032234 032701 001000      13#:
5366 032240 001003
5367 032242 012703 010551
5368 032246 000741
5369 032250 032767 040000 150566 15#:
5370 032256 001003
5371 032260 012703 010526
5372 032264 000732
5373 032266 032701 020000      16#:
5374 032272 001406
5375 032274 012703 010610
5376 032300
      032300 104456
      032302 001301
      032304 000000
      032306 012532
5377 032310 042701 021177      17#:
5378 032314 026727 147756 000001
5379 032322 001404
5380 032324 022701 000200
5381 032330 001411
5382 032332 000402
5383 032334 005701      99#:
5384 032336 001406
5385 032340      18#:
      032340 104456
      032342 001300
      032344 000000
      032346 012766

BR          9#
BIT         @BHSTAT,R1
BNE        10#
MOV        @MBHSTA,R3
ERRHRD     702,,,ERR3
TRAP      C#ERHRD
           702
           0
           0
           ERR3
EXIT      TST
TRAP      C#EXIT
           L10035-.
           MISWIW
           16#
           PASNUM
           16#
           @HSSTAT,R1
           13#
           @MHSTA,R3
           @CCYLUP,R4
           703,,,ERR4
           C#ERHRD
           703
           0
           0
           ERR4
           TST
           C#EXIT
           L10035-.
           @VCSTAT,R1
           15#
           @MVOLCK,R3
           9#
           @DRVERR,T.CS
           16#
           @MDRERR,R3
           9#
           @MLSTAT,R1
           17#
           @MLSTA,R3
           705,,,ERR2
           C#ERHRD
           705
           0
           0
           ERR2
           @21177,R1
           T.DRIVE,#1
           99#
           @200,R1
           19#
           18#
           R1
           19#
           704,,,ERR6
           C#ERHRD
           704
           0
           0
           ERR6

;GO REPORT
;CHECK BRUSH HOME SET
;YES-SKIP
;SET NAME MESSAGE PTR
;REPORT ERROR

;EXIT

;TEST IF MANUAL INTERVENTION RUN
;NO-SKIP
;CHECK IF FIRST PASS
;NO-SKIP
;ELSE CHECK MD 0 SELECTED
;YES-SKIP
;SET NAME MESSAGE PTR
;SET CONDITION POINTER
;REPORT ERROR

;EXIT

;CHECK VOL CHECK SET
;YES-SKIP
;ELSE SET NAME MESSAGE PTR
;GO REPORT
;TEST DRIVE ERROR SET
;YES-SKIP
;ELSE SET NAME MESSAGE PTR
;GO REPORT
;CHECK WRITE LOCK STATUS
;SKIP IF RESET
;ELSE SET NAME MESSAGE PTR

;CLEAR STAU8 EXCEPT FOR ERROR BITS

;NO-SKIP
;ELSE REPORT ALL ERRORS
    
```

*TEST 7 INITIAL STATE

```

5386 032350          EXIT      TST          ;EXIT
      032350 104432  TRAP      C#EXIT
      032352 000016  .WORD    L10035-.
5387 032354 150464 19#:  MOV      T.CS,R1      ;GET CS REG
5388 032360 042701 141777  BIC      @141777,R1  ;CLEAR ALL BUT ERROR BITS
5389 032364 005701  TST      R1          ;TEST IF ANY ERROR SET
5390 032366 001364  BNE      18#         ;YES-SKIP TO REPORT
5391 032370          25#:
5392 032370          100#:
5393 032370          ENDTST
      032370          L10035:
      032370 104401  TRAP      C#ETST

```

```

5394
5395          .SBTTL  *TEST 8          INITIAL RESET STATE
5396 032372          BGNTST          ;TEST 8
      032372          T8::
5397 032372 012767 006515 150412  MOV      @INITST,ERHEAD
5398 032400 004767 164406  JSR      PC,TSTINT  ;INITIALIZE TEST
5399
5400 032404 004767 164420  JSR      PC,GSTATR  ;GET STATUS WITH RESET
5401 032410 032456 65#
5402 032412 005767 161710  TST      MISWIW     ;CHECK IF MAN INTERVENTION WAS RUN
5403 032416 100017 4#          ;NO-SKIP
5404 032420 005767 150734  TST      PASNUM     ;CHECK IF 1ST PASS
5405 032424 001014 4#          ;NO-SKIP
5406 032426 032767 000100 150416  BIT      @HSSTAT,T.MP ;CHECK HD SELECT STILL 0
5407 032434 001410 4#          ;YES-SKIP
5408 032436 012703 010537  MOV      @MHSTA,R3  ;SET NAME MESSAGE PTR
5409 032442 012704 011412  MOV      @CCYLUP,R4 ;SET CONDITION POINTER
5410 032446          ERRHRD 801.,,ERR4 ;REPORT ERROR
      032446 104456  TRAP      C#ERHRD
      032450 001441  .WORD    801
      032452 000000  .WORD    0
      032454 012646  .WORD    ERR4
5411 032456          4#:
5412 032456          65#:
5413 032456          ENDTST
      032456          L10036:
      032456 104401  TRAP      C#ETST

```

```

5414
5415          .SBTTL  *TEST 9          DRIVE READY
5416 032460          BGNTST          ;TEST 9
      032460          T9::
5417 032460 012767 006543 150324  MOV      @T09ERR,ERHEAD ;SET ERROR HEADER
5418 032466 012701 003102  MOV      @NEWCYL,R1  ;GET POINTER TO DESIRED LOC
5419 032472 005021          CLR      (R1).        ;CLEAR NEW CYL
5420 032474 005021          CLR      (R1).        ;CLEAR CURRENT CYL
5421 032476 005021          CLR      (R1).        ;      DIFFERENCE
5422 032500 005011          CLR      (R1)          ;      SIGN
5423 032502 004767 164304  JSR      PC,TSTINT  ;INITIALIZE TEST
5424 032506 004767 164316  JSR      PC,GSTATR  ;GET STATUS WITH RESET
5425 032512 033014 100#
5426 032514 004767 170012  JSR      PC,POSHSB  ;POSITION HEAD SELECTED BIT
5427 032520 010567 150366  MOV      R5,DESHD   ;STORE AS DESIRED HEAD
5428 032524 004767 166346  JSR      PC,SIMSEK  ;EXECUTE SIMPLE SEEK
5429 032530 033014 100#
5430 032532 012703 010404  MOV      @MRDY,R3  ;SET NAME MESSAGE PTR

```

*TEST 9 DRIVE READY

```

5431 032536 012704 011353      MOV      #CDRDY,R4      ;SET CONDITION POINTER
5432 032542 004767 164312      JSR      PC,GSTAT      ;GET STATUS
5433 032546 033014              100$
5434 032550 032767 000001 150266  BIT      #DRDYMSK,T.CS ;TEST READY SET
5435 032556 001406              BEQ      4$             ;NO-SKIP
5436 032560              ERRHRD  901...ERR4     ;REPORT READY ERROR
              032560 104456      TRAP    C#ERHRD
              032562 001605      .WORD  901
              032564 000000      .WORD  0
              032566 012646      .WORD  ERR4
5437 032570              EXIT    TST             ;EXIT
              032570 104432      TRAP    C#EXIT
              032572 000222      .WORD  L10037-.
5438 032574 012701 000121      4$:     MOV      #81.,R1      ;SET WAIT COUNT
5439 032600 004767 164254      5$:     JSR      PC,GSTAT      ;GET STATUS
5440 032604 033014              100$
5441 032606 012703 000005      MOV      #5,R3         ;SET EXPECTED STATE VALUE
5442 032612 026703 150242      CMP     T,STAT,R3     ;CHECK STATE IS 5
5443 032616 001406              BEQ     7$             ;YES-SKIP
5444 032620              ERRHRD  902...ERR7     ;ELSE REPORT
              032620 104456      TRAP    C#ERHRD
              032622 001606      .WORD  902
              032624 000000      .WORD  0
              032626 013666      .WORD  ERR7
5445 032630              EXIT    TST
              032630 104432      TRAP    C#EXIT
              032632 000162      .WORD  L10037-.
5446 032634 012703 010404      7$:     MOV      #MDRDY,R3
5447 032640 032767 000001 150176  BIT      #DRDYMSK,T.CS ;CHECK READY SET
5448 032646 001025              BNE     12$           ;YES-SKIP
5449 032650 005301              DEC     R1             ;ELSE DEC WAIT COUNT
5450 032652 001415              BEQ     9$             ;SKIP IF 0
5451              TIMDLY  1
5452 032654              WAITUS 1               ;JSD REV A
5453 032704 000735              BR      5$             ;JSD REV A
5454 032706              9$:     ERRHRD  903...ERR5 ;REPORT READY ERROR
              032706 104456      TRAP    C#ERHRD
              032710 001607      .WORD  903
              032712 000000      .WORD  0
              032714 012716      .WORD  ERR5
5455 032716              EXIT    TST
              032716 104432      TRAP    C#EXIT
              032720 000074      .WORD  L10037-.
5456
5457 032722 005767 150116      12$:    TST     T.CS          ;TEST IF ANY ERROR
5458 032726 100006              BPL     15$           ;NO-SKIP
5459 032730              ERRHRD  904...ERR6
              032730 104456      TRAP    C#ERHRD
              032732 001610      .WORD  904
              032734 000000      .WORD  0
              032736 012766      .WORD  ERR6
5460 032740              EXIT    TST
              032740 104432      TRAP    C#EXIT
              032742 000052      .WORD  L10037-.
5461 032744 012703 010537      15$:    MOV      #MHSTA,R3     ;SET NAME MESSAGE PTR
5462 032750 004767 167556      JSR     PC,POSHSB     ;POSITION HEAD SELECT BIT FOR TEST
5463 032754 020567 150132      CMP     R5,DESHD     ;CHECK IF CORRECT HEAD SELECTED

```

*TEST 9 DRIVE READY

5464 032760 001415
 5465 032762 005767 150124
 5466 032766 001406
 5467 032770
 032770 104456
 032772 001611
 032774 000000
 032776 012600
 5468 033000
 033000 104432
 033002 000012
 5469 033004
 033004 104456
 033006 001612
 033010 000000
 033012 012532
 5470 033014
 5471 033014
 5472 033014
 033014
 033014 104401

BEQ 20# ;YES-SKIP
 TST DESHD ;ELSE TEST IF 1 DESIRED
 BEQ 17# ;NO-REPORT SB 0
 ERRHRD 905...ERR3 ;ELSE REPORT SB 1
 TRAP C#ERHRD
 .WORD 905
 .WORD 0
 .WORD ERR3
 EXIT TST
 TRAP C#EXIT
 .WORD L10037-.
 17# : ERRHRD 906...ERR2
 TRAP C#ERHRD
 .WORD 906
 .WORD 0
 .WORD ERR2
 20# :
 100# :
 ENDTST
 L10037 :
 TRAP C#ETST

5473
 5474
 5475 033016
 033016
 5476 033016 012767 006553 147766
 5477 033024 012701 003102
 5478 033030 005021
 5479 033032 005021
 5480 033034 005021
 5481 033036 052721 000001
 5482 033042 004767 167464
 5483 033046 010521
 5484 033050
 5485 033050
 033050
 033050 104402

.SBTTL *TEST 10 SEEK SIGN SWITCH
 BGNTST ;TEST 10
 T10 : :
 MOV #T10ERR,ERHEAD ;SET ERROR HEADER
 MOV #NEWCYL,R1
 CLR (R1)+ ;CLEAR NEW CYL
 CLR (R1)+ ;CLEAR CURRENT CYLINDER
 CLR (R1)+ ;CLEAR DIFFERENCE
 BIS #BIT0,(R1)+ ;SET FOR SIGN OF 1
 JSR PC,POSHSB ;GET SELECTED HEAD
 MOV R5,(R1)+ ;SET AS DESIRED HEAD
 T104# :
 BGNSUB
 T10.1 :

5486 033052 004767 163734
 5487 033056 004767 163746
 5488 033062 033354
 5489 033064 004767 166006
 5490 033070 033354
 5491 033072 012703 010404
 5492 033076 012704 011353
 5493 033102 004767 163752
 5494 033106 033354
 5495 033110 032767 000001 147726
 5496 033116 001406
 5497 033120
 033120 104456
 033122 001751
 033124 000000
 033126 012646
 5498 033130
 033130 104432
 033132 000222
 5499

TRAP C#BSUB
 JSR PC,TSTINT ;INITIALIZE TEST
 JSR PC,GSTATR ;GET STATUS
 60#
 JSR PC,SIMSEK ;DO SEEK
 60#
 MOV #MORDY,R3 ;SET NAME MESSAGE PTR
 MOV #CDRDY,R4 ;SET CONDITION MESSAGE PTR
 JSR PC,GSTAT ;GET STATUS
 60#
 BIT #DRDYMSK,T.CS ;CHECK READY RESET
 BEQ 4# ;YES-SKIP
 ERRHRD 1001...ERR4 ;REPORT READY ERROR
 TRAP C#ERHRD
 .WORD 1001
 .WORD 0
 .WORD ERR4
 EXIT SUB ;EXIT SUBTEST
 TRAP C#EXIT
 .WORD L10041-.

*TEST 10 SEEK SIGN SWITCH

```

5500 033134 012701 000121      4#:  MOV      #81.,R1      ;SET WAIT COUNT
5501 033140 004767 163714      5#:  JSR      PC,GSTAT    ;GET STATUS
5502 033144 033354                60#
5503 033146 012703 000005      MOV      #5,R3        ;SET EXPECTED STATE
5504 033152 020367 147702      CMP      R3,T.STAT    ;CHECK STATE IS 5
5505 033156 001406                BEQ      7#           ;YES-SKIP
5506 033160                ERRHRD  1002...,ERR7   ;REPORT STATE ERROR
      033160 104456                TRAP    C#ERHRD
      033162 001752                .WORD  1002
      033164 000000                .WORD  0
      033166 013666                .WORD  ERR7
5507 033170                EXIT    SUB           ;EXIT
      033170 104432                TRAP    C#EXIT
      033172 000162                .WORD  L10041-.
5508 033174 012703 010404      7#:  MOV      #MORDY,R3    ;SET NAME MESSAGE PTR
5509 033200 032767 000001 147636  BIT      #ORDYMSK,T.CS ;CHECK READY SET
5510 033206 001025                BNE     12#         ;YES-SKIP
5511 033210 005301                DEC     R1          ;DO WAIT COUNT
5512 033212 001415                BEQ     9#          ;SKIP IF 0
5513                                ;
5514 033214                TIMDLY 1             ;JSD REV A
5515 033244 000735                WAITUS 1             ;JSD REV A
5516                                BR      5#
5517                                9#:  ERRHRD  1003...,ERR5   ;REPORT READY ERROR
      033246 104456                TRAP    C#ERHRD
      033250 001753                .WORD  1003
      033252 000000                .WORD  0
      033254 012716                .WORD  ERR5
5518 033256                EXIT    SUB           ;EXIT
      033256 104432                TRAP    C#EXIT
      033260 000074                .WORD  L10041-.
5519 033262 005767 147556      12#: TST     T.CS        ;TEST IF ANY OTHER ERROR
5520 033266 100006                BPL     15#         ;NO-SKIP
5521 033270                ERRHRD  1004...,ERR6   ;REPORT ALL ERRORS
      033270 104456                TRAP    C#ERHRD
      033272 001754                .WORD  1004
      033274 000000                .WORD  0
      033276 012766                .WORD  ERR6
5522 033300                EXIT    SUB           ;EXIT
      033300 104432                TRAP    C#EXIT
      033302 000052                .WORD  L10041-.
5523
5524 033304 012703 010537      15#: MOV      #MSTA,R3    ;SET NAME MESSAGE PTR
5525 033310 004767 167216      JSR      PC,POSHSB   ;GET SELECTED HEAD BIT
5526 033314 020567 147572      CMP      R5,DESHD    ;CHECK IF CORRECT
5527 033320 001415                BEQ     20#         ;YES - SKIP
5528 033322 005767 147564      TST     DESHD        ;WAS IT SET
5529 033326 001406                BEQ     17#         ;NO-SKIP
5530 033330                ERRHRD  1005...,ERR3   ;REPORT SB 1
      033330 104456                TRAP    C#ERHRD
      033332 001755                .WORD  1005
      033334 000000                .WORD  0
      033336 012600                .WORD  ERR3
5531 033340                EXIT    SUB
      033340 104432                TRAP    C#EXIT
      033342 000012                .WORD  L10041-.
5532 033344                17#: ERRHRD  1006...,ERR2   ;REPORT SB 0

```

*TEST 10 SEEK SIGN SWITCH

```

033344 104456          TRAP  C#ERHRD
033346 001756          .WORD 1006
033350 000000          .WORD 0
033352 012532          .WORD ERR2

5533
5534 033354          204:
5535 033354          604:
5536 033354          ENDSUB
033354          L10041:
033354 104403          TRAP  C#ESUB
5537 033356 005767 147526  TST  DESSGN          ;CHECK IF BOTH SIGN USED
5538 033362 001404          BEQ  254           ;YES-SKIP
5539 033364 005067 147520  CLR  DESSGN          ;SET FOR SIGN OF 0
5540 033370 000167 177454  JMP  T1044         ;DO TEST AGAIN
5541 033374
5542 033374          254:
033374          ENDTST
033374 104401          L10040:
5543          TRAP  C#ETST
5544          .SBTTL *TEST 11          HEAD ALIGNMENT SUPPORT
5545 033376          BGNTST          ;TEST 11

```

```

5546 033376 032767 000010 160722  BIT  #MDALIGN,MISWIW ;CHECK IF RUN HEAD ALIGNMENT
5547 033404 001411          BEQ  14           ;NO-EXIT
5548 033406 005767 147746          TST  PASNUM        ;TEST IF PASS 0
5549 033412 001006          BNE  14           ;NO-EXIT
5550 033414 026767 147412 147366  CMP  RLDRV,HADONE  ;TEST IF HEAD ALIGN DONE THIS DRIVE
5551 033422 001004          BNE  24           ;NO - SKIP
5552 033424 000167 000422          JMP  T1154        ;GO CHECK WRITE LOCK
5553 033430          14:          EXIT  TST
033430 104432          TRAP  C#EXIT
033432 000520          .WORD L10042-
5554 033434 016767 147372 147346  24:          MOV  RLDRV,HADONE  ;SET HEAD ALIGN DONE FLAG
5555          ;TELL DRV AND CNTRL HD ALIGN TO BE DONE ON
5556 033442          PRINTF #FMT5,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
033442 005046          CLR  -(SP)
033444 156716 147363          BISB RLDRV+1,(SP)
033450 012746 006053          MOV  #DRVNAM,-(SP)
033454 016746 147346          MOV  RLBAS,-(SP)
033460 012746 006042          MOV  #BASADD,-(SP)
033464 012746 011567          MOV  #FMT5,-(SP)
033470 012746 000005          MOV  #5,-(SP)
033474 010600          MOV  SP,RO
033476 104417          TRAP  C#PNTF
033500 062706 000014          ADD  #14,SP
5557          ;HD ALIGN. RSETWRT LCK TO SEL HD 0, SET HD 1
5558 033504          PRINTF #FMT9,#HAMES1
033504 012746 007154          MOV  #HAMES1,-(SP)
033510 012746 011753          MOV  #FMT9,-(SP)
033514 012746 000002          MOV  #2,-(SP)
033520 010600          MOV  SP,RO
033522 104417          TRAP  C#PNTF
033524 062706 000006          ADD  #6,SP
5559          ;+C TO RET TO SUPVR CMD MODE
5560 033530          PRINTF #FMT9,#HAMES2
033530 012746 007237          MOV  #HAMES2,-(SP)
033534 012746 011753          MOV  #FMT9,-(SP)

```

*TEST 11 HEAD ALIGNMENT SUPPORT

```

033540 012746 000002      MOV      #2,-(SP)
033544 010600              MOV      SP,RO
033546 104417              TRAP    C#PNTF
033550 062706 000006      ADD      #6,SP
5561
5562 033554              PRINTF  #FMT9,#HAMES3
033554 012746 007343      MOV      #HAMES3,-(SP)
033560 012746 011753      MOV      #FMT9,-(SP)
033564 012746 000002      MOV      #2,-(SP)
033570 010600              MOV      SP,RO
033572 104417              TRAP    C#PNTF
033574 062706 000006      ADD      #6,SP
;IF HD SEL TP (21, 22) DO NOT EXIST
5563
5564
5565 033600              PRINTF  #FMT9,#HAMES4
033600 012746 007406      MOV      #HAMES4,-(SP)
033604 012746 011753      MOV      #FMT9,-(SP)
033610 012746 000002      MOV      #2,-(SP)
033614 010600              MOV      SP,RO
033616 104417              TRAP    C#PNTF
033620 062706 000006      ADD      #6,SP
;JUMPER DRV RDY AND SEEK INCOMPLETE ON DRV
;LOGIC MOD
5566
5567 033624              PRINTF  #FMTOP1,#OPR12A,#OPR1A,#BASADD,RLBAS,#DRVNAM,<B,RLDRV+1>
033624 005046              CLR      -(SP)
033626 156716 147201      BISB    RLDRV+1,(SP)
033632 012746 006053      MOV      #DRVNAM,-(SP)
033636 016746 147164      MOV      RLBAS,-(SP)
033642 012746 006042      MOV      #BASADD,-(SP)
033646 012746 010135      MOV      #OPR1A,-(SP)
033652 012746 010121      MOV      #OPR12A,-(SP)
033656 012746 011442      MOV      #FMTOP1,-(SP)
033662 012746 000007      MOV      #7,-(SP)
033666 010600              MOV      SP,RO
033670 104417              TRAP    C#PNTF
033672 062706 000020      ADD      #20,SP
;SET WRITE LOCK
5568
5569 033676              BGNSUB
033676
033676 104402              TRAP    C#BSUB
5570 033700 004767 163106      JSR      PC,TSTINT
5571 033704 005067 147076      CLR      DONE
;INITIALIZE TEST
;CLEAR DONE
5572
5573 033710 016767 147116 147116      MOV      RLDRV,L.CS
;SET UP FOR GET STATUS
5574 033716 052767 000104 147110      BIS      #GTSTAT,L.CS
5575 033724 012767 000013 147106      MOV      #GETSTAT!DRSET,L.DA
5576
5577 033732 016762 147102 000004      MOV      L.DA,RLDA(R2)
;DO GET STATUS
5578 033740 016762 147070 000000      MOV      L.CS,RLCSR(R2)
5579
5580 033746 005767 147034      TST     DONE
;CHECK IF DONE
5581 033752 001775      BEQ     13#
;NO-GO CLR CONTROLLER
5582
5583 033754 005067 147026      CLR     DONE
5584 033760 012767 000021 147052 20#      MOV     #HDSSEL!MBSETO,L.DA,LOAD FOR HEAD 1
5585 033766 000240      NOP
5586 033770 032767 020000 147054      BIT     #MLSTAT,T.MP
;CHECK IF WRITE LOCK SET
5587 033776 001003      BNE    22#
;YES-SKIP

```

T11.1:

*TEST 11

HEAD ALIGNMENT SUPPORT

```

5588 034000 042767 000020 147032
5589 034006 016767 147020 147020 22$: BIC #HDSEL,L.DA ;ELSE CLEAR TO HEAD 0
5590 034014 052767 000106 147012 MOV RLDRV,L.CS ;LOAD IN DRIVE NUMBER
5591 034022 016762 147012 000004 BIS -SEEK,L.CS ;SET FOR SEEK
5592 034030 016762 147000 000000 MOV L.DA,RLDA(R2) ;LOAD & EXECUTE SEEK
5593 034036 034036 104422 MOV L.CS,RLCSR(R2)
BREAK ;ALLOW OPERATOR TO INTERRUPT PROGRAM TO GET
TRAP C#BRK ;/BACK TO SUPERVISOR COMMAND MODE

5594
5595 034040 005767 146742 30$: TST DONE
5596 034044 001775 BEQ 30$
5597 034046 000716 BR 11$ ;LOOP
5598 034050 59$:
5599 034050 ENDSUB
034050 L10043: TRAP C#ESUB
034050 104403 T115$:
5600 034052 BGNSUB
5601 034052 034052 T11.2:
034052 104402 TRAP C#BSUB
5602 034054 004767 162732 JSR PC,TSTINT ;INITIALIZE TEST
5603 034060 004767 162744 JSR PC,GSTATR ;CLEAR DRIVE
5604 034064 034150 60$
5605 034066 032767 020000 146756 BIT #WLSTAT,T.MP ;CHECK WRITE LOCK RESET
5606 034074 001425 BEQ 19$ ;YES-SKIP
5607 034076 034076 012746 010102 18$: PRINTF #FMT9,#OPR12 ;REQUEST WRITE LOCK RESET
034102 012746 011753 MOV #OPR12,-(SP)
034106 012746 000002 MOV #FMT9,-(SP)
034112 010600 MOV #2,-(SP)
034114 104417 MOV SP,R0
034116 062706 000006 TRAP C#PNTF
5608 034122 005067 150236 ADD #6,SP
5609 034126 034126 104443 CLR OBUFF ;CLEAR FOR RESPONSE
034130 000404 GMANIL OPRO02,OBUFF,1,NO ;GET RESPONSE
034132 004364 TRAP C#GMAN
034134 000120 BR 10000$
034136 007470 .WORD OBUFF
034140 000001 .WORD T#CODE
034142 .WORD OPRO02
034142 10000$: TST OBUFF ;WAS ANSWER YES
5610 034142 005767 150216 BEQ 18$ ;NO-REPEAT REQUEST
5611 034146 001753 19$:
5612 034150 60$:
5613 034150 ENDSUB
5614 034150 L10044: TRAP C#ESUB
034150 104403 20$:
5615 034152 034152 ENDTST
5616 034152 034152 L10042: TRAP C#ETST
034152 104401 .SBTTL *TEST 12 HEAD SWITCHING
5617 5618 BGNTST ;TEST 12
5619 034154 034154 T12::
5620 034154 012767 006573 146630 MOV #T12ERR,ERHEAD ;SET ERROR HEADER
5621 034162 012701 003102 MOV #NEWCYL,R1 ;GET POINTER TO DESIRED LOCATION

```

*TEST 12 HEAD SWITCHING

5622	034166	005021			CLR	(R1)+		;CLEAR NEW CYLINDER
5623	034170	005021			CLR	(R1)+		;CLEAR CURRENT CYL.
5624	034172	005021			CLR	(R1)+		;CLEAR DIFFERENCE
5625	034174	005021			CLR	(R1)+		;CLEAR SIGN
5626	034176	012721	000001		MOV	#1,(R1)+		;SET FOR HEAD 1
5627	034202							
5628	034202							
	034202							
	034202	104402			TRAP	C#BSUB		
5629	034204	004767	162602		JSR	PC,TSTINT		;INITIALIZE TEST
5630	034210	004767	162614		JSR	PC,GSTATR		;GET STATUS WITH RESET
5631	034214	034506			60#			
5632	034216	004767	164654		JSR	PC,SIMSEK		;DO SEEK
5633	034222	034506			60#			
5634	034224	012703	010404		MOV	#MORDY,R3		;SET NAME MESSAGE PTR
5635	034230	012704	011353		MOV	#CDRDY,R4		;SET CONDITION POINTER
5636	034234	004767	162620		JSR	PC,GSTAT		;GET STATUS
5637	034240	034506			60#			
5638	034242	032767	000001	146574	BIT	#DRDYMSK,T.CS		;CHECK IF READY
5639	034250	001406			BEQ	5#		;NO-SKIP
5640	034252				ERRHRD	1201...ERR4		;REPORT READY ERROR
	034252	104456			TRAP	C#ERHRD		
	034254	002261			.WORD	1201		
	034256	000000			.WORD	0		
	034260	012646			.WORD	ERR4		
5641	034262				EXIT	SUB		;EXIT
	034262	104432			TRAP	C#EXIT		
	034264	000222			.WORD	L10046-.		
5642								
5643	034266	012701	000121		5#:	MOV	#81.,R1	;SET WAIT COUNT
5644	034272	004767	162562		6#:	JSR	PC,GSTAT	;GET STATUS
5645	034276	034506			60#			
5646	034300	012703	000005		MOV	#5,R3		;SET EXPECTED STATE VALUE
5647	034304	020367	146550		CMP	R3,T.STAT		;CHECK IF STATE IS 5
5648	034310	001406			BEQ	7#		;YES-SKIP
5649	034312				ERRHRD	1202...ERR7		;REPORT STATE ERROR
	034312	104456			TRAP	C#ERHRD		
	034314	002262			.WORD	1202		
	034316	000000			.WORD	0		
	034320	013666			.WORD	ERR7		
5650	034322				EXIT	SUB		
	034322	104432			TRAP	C#EXIT		
	034324	000162			.WORD	L10046-.		
5651								
5652	034326	012703	010404		7#:	MOV	#MORDY,R3	;SET NAME MESSAGE PTR
5653	034332	032767	000001	146504	BIT	#DRDYMSK,T.CS		;CHECK DRIVE READY
5654	034340	001025			BNE	12#		;YES-SKIP
5655	034342	005301			DEC	R1		;DEC WAIT COUNT
5656	034344	001415			BEQ	9#		;SKIP IF 0
5657					TIMDLY	1		
5658	034346				WAITUS	1		;JSD REV A
5659	034376	000735			BR	6#		;JSD REV A
5660								
5661	034400				9#:	ERRHRD	1203...ERR5	;REPORT READY ERROR
	034400	104456			TRAP	C#ERHRD		
	034402	002263			.WORD	1203		
	034404	000000			.WORD	0		

*TEST 12 HEAD SWITCHING

```

5662 034406 012716 .WORD ERR5
034410 012716 EXIT SUB ;EXIT
034410 104432 TRAP C#EXIT
034412 000074 .WORD L10046-.

5663
5664 034414 005767 146424 12#: TST T.CS ;TEST IF ANY ERROR
5665 034420 100006 BPL 15# ;NO-SKIP
5666 034422 ERRHRD 1204...ERR6 ;REPORT ALL ERRORS
034422 104456 TRAP C#ERHRD
034424 002264 .WORD 1204
034426 000000 .WORD 0
034430 012766 .WORD ERR6
5667 034432 EXIT SUB
034432 104432 TRAP C#EXIT
034434 000052 .WORD L10046-.

5668 034436 012703 010537 15#: MOV #MHSTA,R3 ;SET NAME MESSAGE PTR
5669 034442 004767 166064 JSR PC,POSHSB ;POSITION HEAD SELECT BIT
5670 034446 026705 146440 CMP DESHD,R5 ;CHECK IF CORRECT HEAD SELECTED
5671 034452 001415 BEQ 20# ;YES-SKIP
5672 034454 005767 146432 TST DESHD ;WAS HEAD 0 SELECTED
5673 034460 001406 BEQ 17# ;YES-SKIP
5674 034462 ERRHRD 1205...ERR3 ;REPORT HEAD SB 1
034462 104456 TRAP C#ERHRD
034464 002265 .WORD 1205
034466 000000 .WORD 0
034470 012600 .WORD ERR3
5675 034472 EXIT SUB ;EXIT
034472 104432 TRAP C#EXIT
034474 000012 .WORD L10046-.

5676 034476 17#: ERRHRD 1206...ERR2 ;ELSE REPORT HEAD SB 0
034476 104456 TRAP C#ERHRD
034500 002266 .WORD 1206
034502 000000 .WORD 0
034504 012532 .WORD ERR2

5677
5678 034506 20#:
5679 034506 60#:
5680 034506 ENDSUB
034506 L10046:

5681 034510 005767 146376 TRAP C#ESUB
5682 034514 001404 TST DESHD ;CHECK IF HD 0 WAS DONE
5683 034516 005067 146370 BEQ 25# ;YES-SKIP
5684 034522 000167 177454 CLR DESHD ;ELSE SET TO HEAD 0
5685 034526 JMP T12# ;REDO TEST
5686 034526 25#:
034526 ENDTST
034526 L10045:

5687 TRAP C#ETST
5688 .SBTTL *TEST 13 READ HEADER (PART 1)
5689 034530 BGNTST ;TEST 13

5690 034530 012767 006605 146254 T13:: MOV #T13ERR,ERHEAD ;SET ERROR HEADER
5691 034536 012701 003102 MOV #NEWCYL,R1 ;GET ADDRESS OF DESIRED LOCATIONS
5692 034542 005071 CLR (R1)+ ;CLEAR NEW CYL
5693 034544 005021 CLR (R1)+ ;CLEAR CURRENT CYL
5694 034546 005021 CLR (R1)+ ;CLEAR DIFF

```

*TEST 13

READ HEADER (PART 1)

```

5695 034550 005021 CLR (R1) ;CLEAR SIGN
5696 034552 005021 CLR (R1) ;CLEAR HEAD
5697 034554
5698 034554 T134: BGNSUB
034554
034554 104402 TRAP C#BSUB T13.1:
5699 034556 004767 162230 JSR PC,TSTINT ;INITIALIZE TEST
5700 034562 004767 162242 JSR PC,GSTATR ;GET STATUS W/RESET
5701 034566 034660 60#
5702 034570 004767 164302 JSR PC,SIMSEK ;DO SEEK
5703 034574 034660 60#
5704 034576 012701 000121 MOV #81.,R1 ;SET WAIT COUNT
5705 034602 004767 165754 JSR PC,RDYWAIT ;WAIT FOR READY
5706 034606 034660 60#
5707
5708 034610 004767 165226 10#: JSR PC,XRDHDC ;DO READ HEADER
5709 034614 034660 60#
5710 034616 012703 010537 MOV #MHSTA,R3 ;SET NAME MESSAGE PTR
5711 034622 004767 165676 JSR PC,POSHW1 ;POSITION HS BIT IN HD WRD 1
5712 034626 020567 146260 CMP R5,DESHD ;CHECK IF HEAD CORRECT
5713 034632 001412 BEQ 15# ;YES-SKIP
5714 034634 ERRHRD 1301...,ERR3 ;REPORT SB 1
034634 104456 TRAP C#ERHRD
034636 002425 .WORD 1301
034640 000000 .WORD 0
034642 012600 .WORD ERR3
5715 034644 EXIT SUB
034644 104432 TRAP C#EXIT
034646 000012 .WORD L10050-
5716 034650 17#: ERRHRD 1302...,ERR2 ;REPORT SB 0
034650 104456 TRAP C#ERHRD
034652 002426 .WORD 1302
034654 000000 .WORD 0
034656 012532 .WORD ERR2
5717
5718 034660 15#:
5719 034660 60#:
5720 034660 ENDSUB
034660 L10050:
034660 104403 TRAP C#ESUB
5721 034662 005767 146224 TST DESHD ;TEST IF HEAD 1 DONE
5722 034666 001007 BNE 20# ;YES-SKIP
5723 034670 012767 000001 146214 MOV #1,DESHD ;ELSE SET TO HEAD 1
5724 034676 016767 146150 146212 MOV HDWRD1,TEMPO ;STORE HDR WORD 1
5725 034704 000723 BR T134# ;DO TEST AGAIN
5726 034706 042767 000177 146202 20#: BIC #177,TEMPO ;CLEAR ALL BUT CYLINDER IN 1ST HEADER
5727 034714 042767 000177 146130 BIC #177,HDWRD1 ;CLEAR ALL BY CYL IN 2ND HEADER
5728 034722 026767 146170 146122 CMP TEMPO,HDWRD1 ;COMPARE IF EQUAL
5729 034730 001406 BEQ 22# ;YES-SKIP
5730 034732 012703 007070 MOV #CYLPER,R3 ;SET NAME MESSAGE PTR
5731 034736 ERRHRD 1306...,ERR1 ;REPORT HEAD ALIGNMENT PROBLEM
034736 104456 TRAP C#ERHRD
034740 002432 .WORD 1306
034742 000000 .WORD 0
034744 012464 .WORD ERR1
5732 034746 22#:
5733 034746 ENDTST

```

*TEST 13

READ HEADER (PART 1)

```

034746          L10047:
034746 104401   TRAP    C#ETST
5734
5735          .SBTTL  *TEST 14      READ HEADER (PART 2)
5736 034750     BGNTST ;TEST 14
034750
5737 034750 012767 006621 146034   MOV    #T14ERR,ERHEAD ;SET ERROR HEADER
5738 034756 012701 003104          MOV    #CURCYL,R1    ;GET ADDRESS OF DESIRED VALUE
5739 034762 005021          CLR    (R1)         ;CLEAR CURRENT CYL
5740 034764 005021          CLR    (R1)         ;CLEAR DESIRED DIFF
5741 034766 005021          CLR    (R1)         ;CLEAR SIGN
5742 034770 005021          CLR    (R1)         ;CLEAR DESIRED HEAD
5743 034772
5744 034772     T153#:
034772     BGNSUB
034772          TRAP    C#BSUB          T14.1:
034772 104402   JSR    PC,TSTINT    ;INITIALIZE TEST
5745 034774 004767 162012   JSR    PC,GSTATR    ;CLEAR DRIVE
5746 035000 004767 162024
5747 035004 035204
5748 035006 004767 164064   JSR    PC,SIMSEK    ;DO SEEK
5749 035012 035204
5750 035014 012701 000310   MOV    #200.,R1     ;SET WAIT COUNT FOR 20 MS
5751 035020 004767 165536   JSR    PC,RDYWAIT   ;WAIT FOR READY
5752 035024 035204
5753 035026 004767 166212   JSR    PC,RDALHD    ;DO READ HEADER ALL HEADERS
5754 035032 035204
5755 035034 005067 145754   CLR    MORECE       ;CLEAR MORE COMPARE ERRORS FOR REPORT
5756 035040 052767 000002 145736   BIS    #HDRCMP,OPFLAG ;SET HDR COMPARE FLAG
5757 035046 005003          CLR    R3           ;CLEAR FOR HDR COUNT
5758 035050 012704 003764   MOV    #IBUFF,R4    ;GET POINTER FOR HDR TO BE CHECKED
5759 035054 012705 003116   MOV    #TEMPO,R5    ;GET POINTER TO TEST AREA
5760 035060 012701 000050   MOV    #40.,R1      ;SET HDR COUNT
5761 035064 011415          MOV    (R4),(R5)    ;GET FIRST HEADER WORD
5762
5763 035066 042715 000100   BIC    #MDHSEL,(R5) ;TEST IF HD 0 DESIRED
5764 035072 005767 146014   TST    DESHD        ;YES-SKIP
5765 035076 001404          BEQ    10#          ;ELSE SET HEAD BIT
5766 035100 052715 000100   BIS    #MDHSEL,(R5) ;CLEAR 2ND WORD OF TEST AREA
5767 035104 005065 000002   CLR    2(R5)        ;COMPARE HEADER WORD
5768 035110 021524          10#: CMP    (R5),(R4)    ;SKIP IF OK
5769 035112 001406          BEQ    13#          ;ELSE POSITION R4 TO BAD WORD
5770 035114 005744          TST    -(R4)        ;REPORT ERROR
5771 035116          ERRHRD 1501...ERR10
035116 104456   TRAP    C#ERHRD
035120 002735   .WORD  1501
035122 000000   .WORD  0
035124 014076   .WORD  ERR10
5772 035126 005724          TST    (R4)         ;BUMP R4 TO NEXT WORD
5773 035130 005203          13#: INC    R3           ;BUMP WORD COUNT
5774 035132 005724          TST    (R4)         ;TEST 2ND WORD IS 0
5775 035134 001406          BEQ    15#          ;YES - SKIP
5776 035136 022544          CMP    (R5),-(R4)   ;POSITION PTRS FOR REPORT
5777 035140          ERRHRD 1501...ERR10 ;REPORT ERROR
035140 104456   TRAP    C#ERHRD
035142 002735   .WORD  1501
035144 000000   .WORD  0
035146 014076   .WORD  ERR10
  
```


*TEST 14

READ HEADER (PART 2)

```

5778 035150 024524          CMP      -(R5),(R4)      ;REPOSITION POINTER
5779 035152 005724          TST      (R4)           ;POSITION R4 PAST ECC WORD
5780 035154 005203          INC      R3             ;BUMP WORD COUNT
5781 035156 005215          INC      (R5)           ;BUMP SECTOR COUNT
5782 035160 011500          MOV      (R5),R0        ;CHECK IF SECTOR IS PAST LAST SECTOR
5783 035162 042700 177700      BIC      @1CHDSEC,R0
5784 035166 022700 000050      CMP      @40.,R0
5785 035172 001002          BNE     17#             ;NO-SKIP
5786 035174 042715 000077      BIC      @HDSEC,(R5)    ;ELSE CLEAR SECTOR TO 0
5787 035200 005301          DEC     R1             ;DEC HDR COUNT
5788 035202 001342          BNE     10#             ;YES-SKIP
5789
5790 035204          60#:
5791 035204          ENDSUB
035204          L10052:
035204 104403          TRAP    C#ESUB
5792 035206 005767 145700      TST     DESHD           ;CHECK IF HD 1 TESTED
5793 035212 001005          BNE     20#             ;YES-SKIP
5794 035214 012767 000001 145670      MOV     @1,DESHD        ;ELSE SET TO HEAD 1
5795 035222 000167 177544          JMP     T153#           ;REDO TEST
5796 035226          20#:
5797 035226          ENDTST
035226          L10051:
035226 104401          TRAP    C#ETST
5798
5799          .SBTTL *TEST 15      DIFFERENCE OF 1 SEEK (PART 1)
5800 035230          BGNTST ;TEST 15
035230
5801
5802 035230 012767 006645 145554      MOV     @P2T01E,ERHEAD ;SET ERROR HEADER
5803 035236 012767 000004 145652      MOV     @4,TEMPO        ;SET PASS COUNT
5804 035244 004767 161542          JSR     PC,TSTINT       ;INITIALIZE TEST
5805 035250 004767 161554          JSR     PC,GSTATR       ;GET STATUS
5806 035254 035642          T1765#
5807 035256 022767 000001 145012      CMP     @1,T.DRIVE      ;RL01 OR RL02?
5808 035264 001404          BEQ     2#              ;BRANCH TO SET UP DIFF ARGUMENT FOR RL01
5809 035266 012767 177776 145626      MOV     @-2,TEMP2       ;ELSE, SET -2 INTO DIFF ARGUMENT FOR RL02
5810
5811 035274 000403          BR      5#              ;/(RL02 HAS DOUBLE THE TRACK DENSITY OF RL01)
5812 035276 012767 177777 145616      2#:  MOV     @-1,TEMP2    ;SET -1 INTO DIFF ARGUMENT FOR -1 SEEK
5813 035304 012704 003104          5#:  MOV     @CURCYL,R4      ;SET POINTERS
5814 035310 012705 003102          MOV     @NEWCYL,R5
5815 035314 004767 164412          JSR     PC,CHOSH0       ;GO CHOOSE HEAD
5816 035320          T172#:
5817 035320          BGNSUB
035320
5818 035322 004767 165570      TRAP    C#BSUB
5819 035326 035576          JSR     PC,GETPOS       ;GET POSITION
5820 035330          60#
035330 104420          INLOOP ;CHECK IF IN ERROR LOOP
5821 035332          TRAP    C#INLP
035332 103005          BNCOMPLETE 3#         ;NO - SKIP
5822 035334 021415          BCC     3#
5823 035336 001005          CMP     (R4),(R5)      ;CHECK IF CURRENT = NEW
5824 035340 004767 164452          BNE     4#              ;NO - SKIP
5825 035344 000441          JSR     PC,ONSWAP       ;ELSE SWAP OLD AND NEW
000441          BR      9#              ;SKIP TO SEEK

```

*TEST 15 DIFFERENCE OF 1 SEEK (PART 1)

5826	035346	005467	145550		3#:	NEG	TEMP2		;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
5827	035352	011415			4#:	MOV	(R4),(R5)		;MOVE CURRENT INTO OLD
5828	035354	026714	144722			CMP	HLMTW,(R4)		;CHECK IF CURRENT AT 255
5829	035360	001014				BNE	7#		;NO - SKIP
5830	035362	022767	000001	144706		CMP	#1,T.DRIVE		;RL01 OR RL02?
5831	035370	001404				BEQ	6#		;BRANCH IF RL01
5832	035372	012767	177776	145522		MOV	#-2,TEMP2		;ELSE, SET UP DIFF ARGUMENT FOR RL02
5833	035400	000421				BR	8#		
5834	035402	012767	177777	145512	6#:	MOV	#-1,TEMP2		;AT MAX CYL, MAKE NEXT SEEK REV
5835	035410	000415				BR	8#		;SKIP
5836	035412	005714			7#:	TST	(R4)		;TEST IF CURRENT AT 0
5837	035414	001013				BNE	8#		;NO - SKIP
5838	035416	022767	000001	144652		CMP	#1,T.DRIVE		;RL01 OR RL02?
5839	035424	001404				BEQ	11#		;BRANCH IF RL01
5840	035426	012767	000002	145466		MOV	#2,TEMP2		;ELSE, SET UP DIFF ARGUMENT FOR RL02
5841	035434	000403				BR	8#		
5842	035436	012767	000001	145456	11#:	MOV	#1,TEMP2		;AT CYL 0, MAKE NEXT SEEK FWRD
5843	035444	066715	145452		8#:	ADD	TEMP2,(R5)		;ADD DIFF TO NEW CYL (+1 OR -1 FOR RL01, ;/+2 OR -2 FOR RL02)
5844									;DO SEEK
5845	035450	004767	162632		9#:	JSR	PC,XSEEK		
5846	035454	035576				60#			
5847	035456	004767	162444			JSR	PC,GDRSTA		;GET DRIVE STATE
5848									
5849	035462	012703	000004			MOV	#4,R3		;SET EXPECTED STATE
5850	035466	020367	145366			CMP	R3,T.STAT		;CHECK DRIVE STATE
5851	035472	001405				BEQ	10#		;YES-SKIP
5852	035474					ERRHRD	101...ERR7		;REPORT STATE ERROR
	035474	104456				TRAP	C#ERRHRD		
	035476	000145				.WORD	101		
	035500	000000				.WORD	0		
	035502	013666				.WORD	ERR7		
5853	035504	000427				BR	16#		;EXIT TEST
5854	035506	012703	000005		10#:	MOV	#5,R3		;SET EXPECTED STATE
5855						TIMELY	50.		;WAIT 5 MS FOR DRIVE STATE CHANGE FROM 4 TO 5 ;JSD REV A
5856	035512					WAITUS	50.		;WAIT 5 MS FOR DRIVE STATE CHANGE FROM 4 TO 5 ;JSD REV A
5857	035542	004767	162360		12#:	JSR	PC,GDRSTA		;GET DRIVE STATE
5858	035546	020367	145306			CMP	R3,T.STAT		;IS STATE 5?
5859	035552	001404				BEQ	16#		;YES-SKIP
5860	035554				14#:	ERRHRD	102...ERR7		;REPORT STATE ERROR
	035554	104456				TRAP	C#ERRHRD		
	035556	000146				.WORD	102		
	035560	000000				.WORD	0		
	035562	013666				.WORD	ERR7		
5861	035564	012701	000062		16#:	MOV	#50,R1		;INITIALIZE WAIT COUNT
5862	035570	004767	164766			JSR	PC,RDYWAIT		;GO WAIT FOR DRIVE READY
5863	035574	035576				60#			
5864	035576	012767	000002	145212	60#:	MOV	#2,ERRSWI		;INIT ERROR SWITCH
5865	035604				ENDSUB				
	035604				L10054:				
5866	035606	104403				TRAP	C#ESUB		
	035606	104410				ESCAPE	TST		;EXIT TEST IF ERROR
	035610	000032				TRAP	C#ESCAPE		
	035612	005367	145300			.WORD	L10053-		
5867	035612	001411				DEC	TEMPO		;DEC PASS COUNT
5868	035616					BEQ	24#		;SKIP IF 0-DONE
5869									
5870	035620	032767	000001	145270		BIT	#BITO,TEMPO		;TEST IF PASS=2

*TEST 15 DIFFERENCE OF 1 SEEK (PART 1)

5871 035626 001003
 5872 035630 004767 164122
 5873 035634 035642
 5874 035636 000167 177456
 5875 035642
 5876 035642
 5877 035642
 035642
 035642 104401

23:
 24:
 T1765:
 ENDTST
 L10053:

BNE 23:
 JSR PC,SWAPHD
 24:
 JMP T172:
 TRAP C#ETST

;NO-SKIP
 ;GO SWAP TO HEAD 1 OR END TEST
 ;ABORT RETURN

5878
 5879

.SBTTL
 BGNTST

*TEST 16

DIFFERENCE OF 1 SEEK (PART 2)
;TEST 16

5880 035644
 035644
 5881 035644 012767 006645 145140
 5882 035652 012767 000004 145236
 5883 035660 004767 161126
 5884 035664 004767 161140
 5885 035670 036134
 5886 035672 004767 164034
 5887 035676 012767 177777 145216
 5888 035704 012703 003102
 5889 035710 012704 003104
 5890 035714 012705 003100
 5891 035720
 5892 035720

T187:
 BGNSUB

MOV #P2T02E,ERHEAD
 MOV #4,TEMPO
 JSR PC,TSTINT
 JSR PC,GSTATR
 T1865:
 JSR PC,CHOSHD
 MOV #-1,TEMP2
 MOV #NEWCYL,R3
 MOV #CURCYL,R4
 MOV #OLDCYL,R5

;SET ERROR HEADER
 ;SET PASS COUNT
 ;INITIALIZE TEST
 ;GET STATUS, CLEAR DRIVE
 ;GO CHOOSE HEAD
 ;SET DIFF ARGUMENT TO -1 (REVERSE)
 ;GET ADDRESSES

T16::

5893 035720 104402
 035720 004767 165170
 5894 035726 036072
 5895 035730
 035730 104420

TRAP C#BSUB
 JSR PC,GETPOS

;GET CURRENT POSITION

T16.1:

5896 035732 103005
 035732 021413
 5897 035734 001005
 5898 035736 004767 164052
 5899 035740 000421
 5900 035744 005467 145150
 5901 035746 011413
 5902 035752 026714 144322
 5903 035754 001004
 5904 035760 012767 177777 145132
 5905 035762 000405
 5906 035770 005714
 5907 035772 001003
 5908 035774 012767 000001 145116
 5909 035776 066713 145112
 5910 036004 004767 162272
 5911 036010 036072
 5912 036014 012701 000226
 5913 036016 004767 164534
 5914 036022 036072
 5915 036026 004767 165062
 5916 036030 036072
 5917 036034 011501
 5918 036036 011501
 5919 036040 161401
 5920 036042 005767 145042

3:
 4:
 7:
 8:
 9:

INLOOP
 TRAP C#INLP
 BNCOMPLETE 3:
 BCC 3:
 CMP (R4),(R3)
 BNE 4:
 JSR PC,ONSWAP
 BR 9:
 NEG TEMP2
 MOV (R4),(R3)
 CMP HLMTW,(R4)
 BNE 7:
 MOV #-1,TEMP2
 BR 8:
 TST (R4)
 BNE 8:
 MOV #1,TEMP2
 ADD TEMP2,(R3)
 JSR PC,XSEEK
 60:
 MOV #150.,R1
 JSR PC,RDYWAIT
 60:
 JSR PC,GETPOS
 60:
 MOV (R5),R1
 SUB (R4),R1
 TST DESSGN

;CHECK IF IN ERROR LOOP
 ;NO - SKIP
 ;CHECK IF CURRENT = NEW
 ;NO - SKIP
 ;ELSE SWAP OLD AND NEW
 ;SKIP TO SEEK
 ;CHANGE DIFF ARGUMENT FOR OPPOSITE DIR
 ;MOV CURRENT INTO NEW
 ;CHECK IF CURRENT AT 255
 ;NO - SKIP
 ;AT MAX CYL, MAKE NEXT SEEK REV
 ;SKIP
 ;TEST IF CURRENT AT 0
 ;NO - SKIP
 ;AT CYL 0, MAKE NEXT SEEK FWRD
 ;ADD DIFF TO NEW CYL (+1 OR -1)
 ;DO SEEK
 ;SET WAIT COUNT FOR 15 MS
 ;WAIT FOR READY
 ;STORE POSITION
 ;GET OLD POSITION
 ;SUBTRACT FROM NEW POINTER (FORWARD)
 ;CHECK IF SIGN FORWARD

*TEST 16

DIFFERENCE OF 1 SEEK (PART 2)

```

5921 036046 001402          BEQ      10$          ;YES-SKIP, ELSE SUB FOR SEEK REVERSE
5922 036050 011401          MOV      (R4),R1     ;GET NEW CYLINDER
5923 036052 161501          SUB      (R5),R1     ;SUBTRACT FROM OLD CYL
5924 036054 022701 000001 10$:  CMP      #1,R1       ;CHECK IF RESULT IS DIFFERENCE OF 1
5925 036060 001404          BEQ      12$          ;YES-SKIP
5926 036062          ERRHRD 201...ERR8 ;ELSE REPORT ERROR
      036062 104456          TRAP    C#ERRHD
      036064 000311          .WORD  201
      036066 000000          .WORD  0
      036070 013736          .WORD  ERR8
5927 036072          12$:
5928 036072 012767 000002 144716 60$:  MOV      #2,ERRSWI   ;INIT ERROR SWITCH
5929 036100          ENDSUB
      036100          L10056:
5930 036102          TRAP    C#ESUB      ;EXIT TEST IF ERROR
      036102 104410          ESCAPE  TST
      036104 000030          TRAP    C#ESCAPE
      036106 005367 145004          .WORD  L10055-.
5931 036106 005367 145004          DEC     TEMPO        ;DEC PASS COUNT
5932 036112 001410          BEQ     30$          ;EXIT IF DONE
5933
5934 036114 032767 000001 144774          BIT     #BIT0,TEMPO ;TEST IF PASS 1 OR 3
5935 036122 001003          BNE     20$          ;YES-SKIP
5936 036124 004767 163626          JSR     PC,SWAPHD   ;GO SWAP TO HEAD 1 OR END TEST
5937 036130 036134          30$:  BR      T187$       ;ABORT RETURN
5938 036132 000672          20$:  BR      T187$       ;LOOP
5939 036134          30$:
5940 036134          T1865$:
5941 036134          ENDTST
      036134          L10055:
5942 036134 104401          TRAP    C#ETST
      036136          ENDMOD
5943
5944          .SBTTL  PARAMETER CODING
5945 036136          BGNMOD  HRDPRM
5946 036136          BGNHRD
      036136 000030          .WORD  L10057-L#HARD/2
5947
5948 036140          GPRML  CNTYPE,CNT,1,YES
      036140 005130          .WORD  T#CODE
      036142 036304          .WORD  CNTYPE
      036144 000001          .WORD  1
5949
5950 036146          GPRMA  CSRMSG,CSR,0,160000,177776,YES
      036146 000031          .WORD  T#CODE
      036150 036220          .WORD  CSRMSG
      036152 160000          .WORD  T#LOLIM
      036154 177776          .WORD  T#HILIM
5951
5952 036156          GPRMA  VECMSG,VECT,0,0,776,YES
      036156 001031          .WORD  T#CODE
      036160 036234          .WORD  VECMSG
      036162 000000          .WORD  T#LOLIM
      036164 000776          .WORD  T#HILIM
5953
5954 036166          GPRMD  DRMSG,DRSB,0,3400,0,7,YES
      036166 004032          .WORD  T#CODE

```

PARAMETER CODING

	036170	036276				.WORD	DRMSG
	036172	003400				.WORD	3400
	036174	000000				.WORD	T#LOLIM
	036176	000007				.WORD	T#HILIM
5955							
5956	036200				GPRML	DRTYPE, TYPDR, 1, YES	
	036200	003130				.WORD	T#CODE
	036202	036254				.WORD	DRTYPE
	036204	000001				.WORD	1
5957							
5958	036206				GPRMD	BRMSG, PRIOR, 0, 340, 0, 7, YES	
	036206	002032				.WORD	T#CODE
	036210	036243				.WORD	BRMSG
	036212	000340				.WORD	340
	036214	000000				.WORD	T#LOLIM
	036216	000007				.WORD	T#HILIM
5959							
5960	036220				ENDHRD		
						.EVEN	
	036220				L10057:		
5961						.EVEN	
5962							
5963							
5964	036220	102	125	123	CSRMSG:	.ASCIZ	/BUS ADDRESS/
	036223	040	101	104			
	036226	104	122	105			
	036231	123	123	000			
5965							
5966	036234	126	105	103	VECMMSG:	.ASCIZ	/VECTOR/
	036237	124	117	122			
	036242	000					
5967							
5968	036243	102	122	040	BRMSG:	.ASCIZ	/BR LEVEL/
	036246	114	105	126			
	036251	105	114	000			
5969							
5970	036254	104	122	111	DRTYPE:	.ASCIZ	/DRIVE TYPE = RL01/
	036257	126	105	040			
	036262	124	131	120			
	036265	105	040	075			
	036270	040	122	114			
	036273	060	061	000			
5971							
5972	036276	104	122	111	DRMSG:	.ASCIZ	/DRIVE/
	036301	126	105	000			
5973							
5974	036304	122	114	061	CNTYPE:	.ASCIZ	/RL11/
	036307	061	000				
5975							
5976	036311				ENDMOD		
5977							
5978					.EVEN		
5979							
5980	036312				BGNMOD	SFTPRM	
5981	036312				BGNSFT		
	036312	000016				.WORD	L10060-L#SOFT/2
5982							

PARAMETER CODING

5983	036314				GPRML	SELQ,MISWI,4,YES
	036314	000130				.WORD T#CODE
	036316	036350				.WORD SELQ
	036320	000004				.WORD 4
5984						
5985	036322				GPRML	ALGNQ,MISWI,10,YES
	036322	000130				.WORD T#CODE
	036324	036403				.WORD ALGNQ
	036326	000010				.WORD 10
5986						
5987	036330				GPRML	MANQ,MISWI,100000,YES
	036330	000130				.WORD T#CODE
	036332	036442				.WORD MANQ
	036334	100000				.WORD 100000
5988						
5989	036336				3#:	GPRMD ERLIMQ,ERLIM,D,377,0,377,YES
	036336	004052				.WORD T#CODE
	036340	036477				.WORD ERLIMQ
	036342	000377				.WORD 377
	036344	000000				.WORD T#LOLIM
	036346	000377				.WORD T#HILIM
5990						
5991	036350				ENDSFT	
						.EVEN
	036350				L10060:	
5992						.EVEN
5993						
5994						
5995	036350	105	130	105	SELQ:	.ASCIZ /EXECUTE DRIVE SELECT TESTS/
	036353	103	125	124		
	036356	105	040	104		
	036361	122	111	126		
	036364	105	040	123		
	036367	105	114	105		
	036372	103	124	040		
	036375	124	105	123		
	036400	124	123	000		
5996						
5997	036403	105	130	105	ALGNQ:	.ASCIZ /EXECUTE HEAD ALIGNMENT SUPPORT/
	036406	103	125	124		
	036411	105	040	110		
	036414	105	101	104		
	036417	040	101	114		
	036422	111	107	116		
	036425	115	105	116		
	036430	124	040	123		
	036433	125	120	120		
	036436	117	122	124		
	036441	000				
5998						
5999	036442	104	117	040	MANQ:	.ASCIZ /DO MANUAL INTERVENTION TESTS/
	036445	115	101	116		
	036450	125	101	114		
	036453	040	111	116		
	036456	124	105	122		
	036461	126	105	116		
	036464	124	111	117		

PARAMETER CODING

	036467	116	040	124	
	036472	105	123	124	
	036475	123	000		
6000					
6001	036477	111	116	120	ERLIMQ: .ASCIZ /INPUT ERROR LIMIT/
	036502	125	124	040	
	036505	105	122	122	
	036510	117	122	040	
	036513	114	111	115	
	036516	111	124	000	
6002					
6003					.EVEN
6004					
6005	036522				ENDMOD
6006					
6007	036522				LASTAD
	036522	000000			.EVEN
	036524	000000			.WORD 0
	036526				.WORD 0
6008					L#LAST::
6009					.EVEN
6010	036526				L#LAST::
6011					
6012		000001			.END

SYMBOL TABLE

ADR	=	000020	G
ALGNQ	=	036403	
ALLCYL	=	000001	
ALLSEC	=	000002	
ANYERR	=	100000	
ASSEMB	=	000010	
BADADD	=	004000	
BAMSK	=	000060	
BANAM	=	006125	
BASADD	=	006042	
BELL	=	011274	
BHSTAT	=	000010	
BIT0	=	000001	G
BIT00	=	000001	G
BIT01	=	000002	G
BIT02	=	000004	G
BIT03	=	000010	G
BIT04	=	000020	G
BIT05	=	000040	G
BIT06	=	000100	G
BIT07	=	000200	G
BIT08	=	000400	G
BIT09	=	001000	G
BIT1	=	000002	G
BIT10	=	002000	G
BIT11	=	004000	G
BIT12	=	010000	G
BIT13	=	020000	G
BIT14	=	040000	G
BIT15	=	100000	G
BIT2	=	000004	G
BIT3	=	000010	G
BIT4	=	000020	G
BIT5	=	000040	G
BIT6	=	000100	G
BIT7	=	000200	G
BIT8	=	000400	G
BIT9	=	001000	G
BOE	=	000400	G
BRMSG	=	036243	
BSFLAG	=	003020	
BSFVAL	=	003372	
BSNSTR	=	010307	
BYPSPM	=	010240	
CAFDT	=	011423	
CAMSK	=	002312	
CCYLUP	=	011412	
CDRDY	=	011353	
CHOSHD	=	021732	
CKDATA	=	000102	
CKERLM	=	016444	
CLKADR	=	003146	
CLKFLG	=	003144	
CLKINT	=	016430	G
CLNCOD	=	016212	G
CLRBYT	=	002304	
CLRPAR	=	025042	
CNT	=	000012	
CNTYPE	=	036304	
COMPOP	=	007777	
CONING	=	000004	
CONTIN	=	015032	
COSTAT	=	000040	
COUNT	=	003154	
CRDYMS	=	000200	
CSNAM	=	006120	
CSR	=	000000	
CSRMSG	=	036220	
CURCYL	=	003104	
CYLPER	=	007070	
CYL ⁷⁸¹	=	002604	
CYLU	=	000004	
CYLWC	=	010225	
C#AU	=	000052	
C#AUTO	=	000061	
C#BRK	=	000022	
C#BSEG	=	000004	
C#BSUB	=	000002	
C#CEFG	=	000045	
C#CLCK	=	000062	
C#CLEA	=	000012	
C#CLOS	=	000035	
C#CLP1	=	000006	
C#CVEC	=	000036	
C#DCLN	=	000044	
C#DODU	=	000051	
C#DRPT	=	000024	
C#DU	=	000053	
C#EDIT	=	000003	
C#ERDF	=	000055	
C#ERMR	=	000056	
C#ERRO	=	000060	
C#ERSF	=	000054	
C#ERSO	=	000057	
C#ESCA	=	000010	
C#ESEG	=	000005	
C#ESUB	=	000003	
C#ETST	=	000001	
C#EXIT	=	000032	
C#GETB	=	000026	
C#GETW	=	000027	
C#GMAN	=	000043	
C#GPHR	=	000042	
C#GPLO	=	000030	
C#GPRI	=	000040	
C#INIT	=	000011	
C#INLP	=	000020	
C#MANI	=	000050	
C#MEM	=	000031	
C#MSG	=	000023	
C#OPEN	=	000034	
C#PNTB	=	000014	
C#PNTF	=	000017	
C#PNTS	=	000016	
C#PNTX	=	000015	
C#QIO	=	000377	
C#RDBU	=	000007	
C#REFG	=	000047	
C#RESE	=	000033	
C#REVI	=	000003	
C#RFLA	=	000021	
C#RPT	=	000025	
C#SEFG	=	000046	
C#SPRI	=	000041	
C#SVEC	=	000037	
C#TPRI	=	000013	
C10MS	=	011373	
C5SEC	=	011434	
C500MS	=	011404	
DANAM	=	006132	
DATACH	=	000001	
DCKERR	=	004000	
DCLIM	=	000012	
DCLIMW	=	014340	
DESDIF	=	003106	
DESHD	=	003112	
DESSEC	=	003114	
DESSGN	=	003110	
DIAGMC	=	000000	
DIFAUG	=	003076	
DIFWD	=	010201	
DIRBIT	=	000004	
DIRMSK	=	002314	
DLTERR	=	010000	
DLYCNT	=	003142	
DONE	=	003006	
DRDYMS	=	000001	
DRMSG	=	036276	
DRSB	=	000010	
DRSELT	=	000004	
DRSET	=	000010	
DRTYPE	=	036254	
DRVcnt	=	003074	
DRVERR	=	040000	
DRVNAM	=	006053	
DSESTA	=	000400	
DSMSK	=	001400	
DSPCOD	=	014342	G
EF.CON	=	000036	G
EF.NEW	=	000035	G
EF.PWR	=	000034	G
EF.RES	=	000037	G
EF.STA	=	000040	G
ERHEAD	=	003012	
ERLIM	=	000010	
ERLIMQ	=	036477	
ERLIMW	=	014336	
ERRCNT	=	003160	
ERRPOI	=	003156	
ERRSWI	=	003016	
ERRVEC	=	003140	
ERR1	=	012464	G
ERR10	=	014076	G
ERR2	=	012532	G
ERR3	=	012600	G
ERR4	=	012646	G
ERR5	=	012716	G
ERR6	=	012766	G
ERR7	=	013666	G
ERR8	=	013736	G
ERR9	=	014032	G
EVL	=	000004	G
EXT05	=	030604	
E#END	=	002100	
E#LOAD	=	000035	
FBSFIL	=	003570	
FMTOP1	=	011442	
FMTOP2	=	011471	
FMTOP3	=	011513	
FMT1	=	011534	
FMT1.1	=	011541	
FMT11	=	011760	
FMT12	=	011766	
FMT13	=	011774	
FMT14	=	012040	
FMT15	=	012072	
FMT16	=	012126	
FMT17	=	012137	
FMT18	=	012161	
FMT19	=	012213	
FMT2	=	011550	
FMT20	=	012250	
FMT21	=	012300	
FMT22	=	012323	
FMT23	=	012357	
FMT24	=	012373	
FMT25	=	012400	
FMT26	=	012410	
FMT27	=	012434	
FMT28	=	012453	
FMT3	=	011553	
FMT4	=	011556	
FMT5	=	011567	
FMT6	=	011607	
FMT7	=	011651	
FMT8	=	011721	
FMT9	=	011753	
FOLWRT	=	000100	
FRWD	=	010232	
FWDSK0	=	002000	
FWDSKS	=	000400	
F#AU	=	000015	
F#AUTO	=	000020	
F#BGN	=	000040	
F#CLEA	=	000007	
F#DU	=	000016	
F#END	=	000041	
F#HARD	=	000004	
F#HW	=	000013	
F#INIT	=	000006	
F#JMP	=	000050	
F#MOD	=	000000	
F#MSG	=	000011	
F#PROT	=	000021	
F#PWR	=	000017	
F#RPT	=	000012	
F#SEG	=	000003	
F#SOFT	=	000005	
F#SRV	=	000010	
F#SUB	=	000002	
F#SW	=	000014	
F#TEST	=	000001	
GBND	=	002310	
GDRSTA	=	020126	
GETPOS	=	023116	
GETSTA	=	000003	
GLBDAT	=	002224	G
GLBEQA	=	002224	G
GLBERR	=	012464	G
GLBSUB	=	016444	G
GLBXTX	=	005242	G
GSTAT	=	017060	
GSTATC	=	017044	
GSTATG	=	017070	
GSTATR	=	017030	
GSTER1	=	006464	
GTSTAT	=	000104	
G#CNT0	=	000200	
G#DELM	=	000372	
G#DISP	=	000003	
G#EXCP	=	000400	
G#HILI	=	000002	
G#LOLI	=	000001	
G#NO	=	000000	
G#OFFS	=	000400	
G#OFSI	=	000376	
G#PRMA	=	000001	
G#PRMD	=	000002	
G#PRML	=	000000	
G#RADA	=	000140	
G#RADB	=	000000	
G#RADD	=	000040	
G#RADL	=	000120	
G#RADO	=	000020	
G#XFER	=	000004	
G#YES	=	000010	
HADONE	=	003010	
HAMES1	=	007154	
HAMES2	=	007237	
HAMES3	=	007343	
HAMES4	=	007406	
HCESTA	=	040000	
HCR CER	=	004000	
HDALIG	=	000010	
HDCYL	=	002316	

SYMBOL TABLE

HDHSEL = 000100	LAB1 006144	L#SPCP 002020 G	L10057 036220	MRSLT 005423
HDMOVF 007031	LAB2 006157	L#SPTP 002024 G	L10060 036350	MSEEK 005242
HDRCMP = 000002	LBASE 003150	L#STA 002030 G	MANQ 036442	MSPERR 010662
HDR40 = 100000	LCLEXT 031572	L#SW 014326 G	MBADAD 005721	MTERR 010715
HDSEC = 000077	LCLK 014544	L#TEST 002114 G	MBADSF 005742	MTMBS 006020
HDSEL = 000020	LCLK1 014552	L#TIML 002014 G	MBHSTA 010575	MTOSLO 006200
HDWD 010214	LOCERR 003364	L#UNIT 002012 G	MBSETO = 000001	MULOAD 005434
HDWRD1 003052	LOCYL = 040000 G	L.BA 003036	MCERR 010415	MUNDEF 011103
HDWRD2 003054	LOE = 040000 G	L.CS 003034	MCONHN 006273	MVOLCK 010551
HDWRD3 003056	LQIM = 000002	L.DA 003040	MCOSTA 010562	MWDERR 010751
HEAD = 000006	LQIMW 014330	L.MP 003042	MCYLOC 011052	MWGERR 010700
HEADLM = 010000	LOT = 000010 G	L10000 012530	MCYLUP 005445	MWLSTA 010610
HEADW 014334	LPT05 030364	L10001 012576	MDATCP 005327	MWORD 006172
HICYL = 020000	L#ACP 002110 G	L10002 012644	MDCRC 010437	MWRCHK 005271
HILIM = 000004	L#APT 002036 G	L10003 012714	MDHEDR 002000 G	MWRITE 005303
HILIMW 014332	L#AUT 002070 G	L10004 012764	MDLT 010464	MWRSET 005400
HLMTW 002302	L#AUTO 015654 G	L10005 013664	MDRDY 010404	MWRTAB 011211
HNFERR = 010000	L#CCP 002106 G	L10006 013734	MDRERR 010526	M40HDR 005364
HOE = 100000 G	L#CLEA 016212 G	L10007 014030	MDRRES 006220	NEWCYL 003102
MOSTAT = 000020	L#CO 002032 G	L10010 014074	MDRVST 010647	NOCLK 014654
HPTCOD 014306 G	L#DEPO 002011 G	L10011 014304	MDSERR 010632	NOCLR = 000010
HRDPRM 036136 G	L#DESC 002122 G	L10012 014324	MERRS 011265	NOCTLR 006754
HRDWTS 025072 G	L#DESP 002076 G	L10013 014342	MEXERS 011225	NOERCT 003365
HSMASK = 000100	L#DEVP 002060 G	L10015 015652	MFLERR 011013	NOIRPT = 000002
HSSTAT = 000100	L#DISP 014344 G	L10016 016210	MFHTER 005773	NOOP = 000100
IBE = 010000 G	L#DLY 002116 G	L10017 016366	MFOLWR 005521	NOPCLK 014472
IBUFF 003764	L#DTP 002040 G	L10020 016372	MFWSK 005576	NOPIR 006060
IDU = 000040 G	L#DTP 002034 G	L10021 016426	MFWSKO 005631	NOTRDY 007003
IER = 020000 G	L#DU 016370 G	L10022 016434	MGTSTA 005315	NOTST 006664
INITCO 014412 G	L#DUT 002072 G	L10023 016442	MHCERR 010733	NSTACH 006416
INITST 006515	L#DVTY 002212 G	L10024 025350	MHCRC 010427	NXMERR = 020000
INDUTS = 000020	L#EF 002052 G	L10025 025556	MHDERR 010776	NXTHL 002306
INTEBL = 000100	L#ENVI 002044 G	L10026 027276	MHDRCP 005346	NXTPAS 015052
INTHLR 016374 G	L#ETP 002102 G	L10027 030200	MHFRC 010476	OBUFF 004364
ISR = 000100 G	L#EXP1 002046 G	L10030 030076	MHF 010450	OLDCYL 003100
IXE = 004000 G	L#EXP4 002064 G	L10031 030604	MHOSTA 010621	ONSWAP 022016
I#AU = 000041	L#EXP5 002066 G	L10032 030512	MHSTA 010537	OPFLAG 003004
I#AUTO = 000041	L#HARD 036140 G	L10033 031572	MINOUT 005476	OPIERR = 002000
I#CLN = 000041	L#HIME 002120 G	L10034 031516	MISTST 006373	OPMSG 002224
I#DU = 000041	L#HPCP 002016 G	L10035 032370	MISWI = 000000	OPR002 007470
I#HRD = 000041	L#HPTP 002022 G	L10036 032456	MISWIW 014326	OPR003 007515
I#INIT = 000041	L#HM 014310 G	L10037 033014	MITEST = 100000	OPR004 010164
I#MOD = 000041	L#ICP 002104 G	L10040 033374	MINDRST 011057	OPR1 007540
I#MSG = 000041	L#INIT 014412 G	L10041 033354	MNEERR 011041	OPR1A 010135
I#PROT = 000040	L#LADP 002026 G	L10042 034152	MNOCLR 006307	OPR1B 010141
I#PTAB = 000041	L#LAST 036526 G	L10043 034050	MNOINT 006240	OPR10 010003
I#PMR = 000041	L#LOAD 002100 G	L10044 034150	MOPER 005414	OPR11 010051
I#RPT = 000041	L#LUN 002074 G	L10045 034526	MOPERR 010766	OPR12 010102
I#SEG = 000041	L#MREV 002050 G	L10046 034506	MORECE 003014	OPR12A 010121
I#SETU = 000041	L#NAME 002000 G	L10047 034746	MOUTIN 005455	OPR2 007616
I#SFT = 000041	L#PRIO 002042 G	L10050 034660	MPNAM 006137	OPR3 007651
I#SRV = 000041	L#PROT 014404 G	L10051 035226	MQUALS = 003760	OPR6 007665
I#SUB = 000041	L#PRT 002112 G	L10052 035204	MREAD 005250	OPR7 007720
I#TST = 000041	L#REPP 002062 G	L10053 035642	MREADH 005261	OPR8 007747
JJJ 002300	L#REV 002010 G	L10054 035604	MRESKO 005665	OPR9 007766
J#JMP = 000167	L#SOFT 036314 G	L10055 036134	MREVSK 005543	OUTINS = 000040
LAB 015004	L#SPC 002056 G	L10056 036100	MRLFAL 011150	O#APTS = 000000

SYMBOL TABLE

O#AU = 000000	READRL 016604	SVCSUB= 000001	T#TSTM= 177777	T16.1 035720
O#BGNR= 000000	RELDWT= 040000	SVCTAG= 000000	T#TSTS= 000001	T172# 035320
O#BGNS= 000001	RESE3 011300	SVCTST= 000001	T#AUT= 010016	T1765# 035642
O#DU = 000001	RESE4 011304	SWAPHD 021756	T#CLE= 010017	T1865# 036134
O#ERRT= 000000	RESE5 011311	S#LSYM= 010000	T#DU = 010020	T187# 035720
O#GNSW= 000001	RESE6 011316	TBLSTR 003024	T#HAR= 010057	T2 025352 G
O#POIN= 000001	RESPAR 003062	TBT 002544	T#HW = 010012	T25TBL 002430
O#SETU= 000000	RESTAR 015022	TCERR 010363	T#INI= 010015	T25TB2 002456
PART1 = 000001 G	RESTBL 002320	TCLK 014614	T#MSG= 010011	T3 025560 G
PASCNT 003152	REVSKO= 001000	TEMPO 003116	T#PRO= 010014	T33TBL 002504
PASNEW 015060	REVSKS= 000200	TEMP1 003120	T#SQF= 010060	T365# 027276
PASNUM 003360	RLBA = 000002	TEMP2 003122	T#SRV= 010023	T4 027300 G
PATTBL 002360	RLBAS 003026	TEMP3 003124	T#SUB= 010056	T4.1 027320
PAT1 004764	RLCS = 000000	TEMP4 003126	T#SW = 010013	T465# 030070
PAT10 005240	RLCSR = 000000	TEMP5 003130	T#TES= 010055	T5 030202 G
PAT2 004766	RLDA = 000004	TEMP6 003132	T.BA 003046	T5.1 030474
PAT3 005026	RLDRV 003032	TEMP7 003134	T.CS 003044	T504# 030526
PAT4 005066	RLMP = 000006	TEMP8 003136	T.DA 003050	T6 030606 G
PAT5 005126	RLVEC 003030	TOSLOW= 000001	T.DRIV 002276	T6.1 031224
PAT6 005134	RORWOP= 020000	TRPFLG 003366	T.MP 003052	T7 031574 G
PAT7 005174	RPTOP 023612	TRPHAN 016436 G	T.STAT 003060	T8 032372 G
PAT8 005176	RPTREM 024606	TSTCLK 014606	TOSERR 006530	T9 032460 G
PAT9 005236	RPTRES 024400	TSTINT 017012	T09ERR 006543	UAM = 000200 G
PCLK 014440	RSTRT 014740	TSTLAB 006365	T1 025072 G	ULOAD = 000010
PNT = 001000 G	SAMSK = 000077	TST4 027322	T10 033016 G	UNDTST 010151
POSHDO 022536	SBSFIL 003374	TYPDR = 000006	T10ERR 006553	UNXERR 006350
POSHSB 022532	SECWD 010220	T#ARGC= 000002	T10.1 033050	VCNRST 006327
POSHW1 022524	SEEK = 000106	T#CODE= 004052	T104# 033050	VCSTAT= 001000
PRI = 002000 G	SEEKOP= 010000	T#ERRN= 000311	T11 033376 G	VECMG 036234
PRIOR = 000004	SELQ 036350	T#EXCP= 000000	T11.1 033676	VECT = 000002
PRI00 = 000000 G	SEQMES 010253	T#FLAG= 000040	T11.2 034052	WAITIN 016636
PRI01 = 000040 G	SETDON 015106	T#GMAN= 000000	T115# 034052	WCMSK = 017777
PRI02 = 000100 G	SFTPRM 036312 G	T#HILI= 000377	T12 034154 G	WCRNG = 160000
PRI03 = 000140 G	SGNMD 010207	T#LAST= 000001	T12ERR 006573	WDESTA= 100000
PRI04 = 000200 G	SIMSEK 021076	T#LOLI= 000000	T12.1 034202	WGESTA= 002000
PRI05 = 000240 G	SPDERR 006430	T#LSYM= 010000	T124# 034202	WLSTAT= 020000
PRI06 = 000300 G	SPDSTA= 004000	T#LTND= 000020	T13 034530 G	WRTSWI 003022
PRI07 = 000340 G	SPTCOD 014324 G	T#NEST= 177777	T13ERR 006605	WTDATA= 000112
PSETNM 003362	SSINDX 003002	T#NSO = 000000	T13.1 034554	XRDHD 022052
PMCON 015330	STAMES 010276	T#NS1 = 000005	T134# 034554	XRDHDC 022042
PWRFLG 003370	STAMSK= 000007	T#NS2 = 000002	T14 034750 G	XRDHDG 022056
P2T01E 006645	STATE2 011323	T#PTNU= 000000	T14ERR 006621	XSEEK 020306
P2T02E 006645	STATE3 011333	T#SAVL= 177777	T14.1 034772	XSEEKT 020276
RDALHD 023244	STATE5 011343	T#SEGL= 177777	T15 035230 G	XSEEK1 020312
RDDATA= 000114	STOSTA= 010000	T#SUBN= 000001	T15.1 035320	X#ALWA= 000000
RDHEAD= 000110	SUBSTK 002404	T#TAGL= 177777	T153# 034772	X#FALS= 000040
RDNDHR= 000116	SVCBGL= 000001	T#TAGN= 010061	T16 035644 G	X#OFFS= 000400
RDYCHK 021406	SVCGBL= 000000	T#TEMP= 000000	T16ERR 006635	X#TRUE= 000020
RDYMAI 022562	SVCINS= 000000	T#TEST= 000020		

. ABS. 036526 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29464 WORDS (116 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:41:04
CNRLIA.BIN,CNRLIA.LST/-SP=SVC34.MLB/ML,CNRLIA.MAC