

11/21+
RL01/02

RL01/2 DRIVE COMPAT
CNRLLA0

COPYRIGHT (c) 1982-83
RH-T754A-MC
FICHE 1 OF 1

APR 1984
Digital
Made In USA

Microfiche grid containing multiple frames of data, likely technical specifications or code, arranged in a grid pattern. The text is too small to read accurately but appears to be organized into columns and rows.

11
07

1
2
3 000000
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
49
50
51

.TITLE CNRLLA0 RL01/02 DRIVE COMPAT
.ENABLE AMA
.ENABLE ABS
.NLIST TOC
.REM @

IDENTIFICATION

PRODUCT CODE: AC-T753A-MC
PRODUCT NAME: CNRLLA0 RL01/2 DRIVE COMPATIBILITY
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

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74

REVISION HISTORY

CHANGES MADE TO CZRLLC0 IN PRODUCING CNRLLA0 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.
2. SET VECTOR 140 WITH THE ADDRESS OF ODT IN ROM (170000).
3. CHANGED REGISTER USAGE NEAR LABEL "CMPENA" AT THE END OF THE INIT CODE, NUMBER OF UNITS LOADED INTO R2 INSTEAD OF R0, WHICH GETS DESTROYED BY DODU MACRO EXPANSION.
4. CHANGED "POINTER" MACRO ARGUMENT FROM "NONE" TO "BGNDU" IN ORDER TO FIX TRAP ERR CAUSED BY A ZERO DROP UNIT ROUTINE ADDRESS IN THE DRS HEADER AREA (LOCATION 2072).

76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

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
3.2	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

121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
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

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+. 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 XXDP+ 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

THE RL01 DRIVE COMPATABILITY TEST IS A KXT-11 (SBC-11/21+) BASED PROGRAM THAT WILL TEST INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES. THE TEST PERFORMS WRITES, READS, OVERWRITES, ADJACENT CYLINDER WRITES TO PROVE COMPATABILITY. SINCE THE PROGRAM RELIES ON MANUAL INTERVENTION, A TOTAL TEST TIME IS NOT APPLICABLE. HOWEVER, TO TEST TWO DRIVES REQUIRES A MINIMUM OF THREE PASS. EACH PASS REQUIRES APPROXIMATELY 70 SECONDS.

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.)

177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227

* 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

CNRLLAO RL01/02 DRIVE COMPATABILITY
(FORMERLY CZRLF8)

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

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

CVRLABO	RLV11 RL01/02 DISKLESS TEST (RLV11 ONLY)
CNRLGAO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 1)
CNRLHAO	RL11/RLV11 RL01/02 CONTROLLER TEST (PART 2)
CNRLIAO	RL01/02 DRIVE TEST (PART 1)
CNRLJAO	RL01/02 DRIVE TEST (PART 2)
CNRLKAO	RL11/RLV11 RL01/02 PERFORMANCE EXERCISER

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 NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279

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 5 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 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 SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS:HOE
```

281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335

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.

* 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.

337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384

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.

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.

386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436

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 OCCURRED. 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 ON ERROR).

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
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.

438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470

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.

472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521

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

	BY WHOM ENTERED: -----
.R NRLLAO	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV. C APR-79	D
CNRL-A-0	D
CNRL VERIFIES INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D.O
CHANGE HW (L) ? Y	D.O
* UNITS (D) ? 2	D.O
UNIT 0	D
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
DRIVE (O) 0 ?	D.O
DRIVE TYPE = RL01 (L) Y ?	D.O
UNIT 1	D
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
DRIVE (O) 0 ? 1	D.O
DRIVE TYPE = RL01 (L) ? N	D.O (N=RL02)
CNRL HRD ERR 00004 TST 003 SUB 002 PC:004130 ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D.O
***** 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	O
DR>CON/FLAGS:HOE:IER:LOE=0	D.O

523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577

```
CNRL EOP 1          D
↑C
DR>RESTART/PASS:1  D,0
-----
-----
-----
-----
```

2.2 CHAIN MODE OPERATION

NOTE THIS PROGRAM IS NOT CHAINABLE. 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.

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 PASS COUNT IS NOT 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 SOFTWARE SWITCH REGISTER 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.

579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

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 FLAGS ZFLAGS EXIT
4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS EXIT

629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680

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:

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, SUBTEST, 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

682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733

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
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.

735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786

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(CCEED)/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:

- 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.

788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839

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)

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 "# 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.

841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
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

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 RL01'S AND THE LAST 4 DRIVES ARE RL02'S (ON THE SECOND CONTROLLER):

UNITS (0) ? 8

UNIT 0
BUS ADDRESS (0) 174400 ?
VECTOR (0) 160 ?
DRIVE (0) 0 ? 0-3
DRIVE TYPE = RL01 (L) Y ?

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

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

THE SECOND TIME THRU THE P-TABLE QUESTIONS THE FIRST 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 #2. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #3 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #4.

890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941

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.

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?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

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.

THERE ARE NO SOFTWARE PARAMETERS.

3.0 ERROR INFORMATION

ERROR INFORMATION IS COMPLETE IN GIVING ALL INFORMATION NECESSARY. ALL REGISTERS ARE GIVEN AS WELL AT TRACK, SECTOR AND DRIVES INVOLVED IN ERROR.

943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
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

3.1 ERROR REPORTING

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DEVICE. ERROR REPORTS ARE AIMED AT BEING SELF EXPLANATORY. THE GENERAL FORMAT IS:

CNRLX XXX ERR YYYYY TST ZZZ SUB PPP PC: RRRRRR

WHERE:

?	IS PROGRAM LETTER
XXX	IS SFT - SOFT ERROR
	HRD - HARD ERROR
	DV FAT - DEVICE FATAL ERROR
	SYS FAT - SYSTEM FATAL ERROR
YYYYY	IS THE ERROR NUMBER
ZZZ	IS THE TEST NUMBER
PPP	IS THE SUBTEST NUMBER
RRRRRR	IS THE PROGRAM LISTING LOCATION

ERRORS GIVE THE REGISTER CONTENTS BEFORE AND AFTER THE ERROR ALONG WITH A ONE LINE DESCRIPTION AND RELEVANT DATA.

EXAMPLE:

ONE LINE DESCRIPTION
(OPTIONAL SECOND LINE)
(OPTIONAL THIRD LINE)
BEFORE CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX
AFTER CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX
OTHER PERTINENT INFORMATION IS GIVEN AT THIS TIME.

REGISTER DESCRIPTIONS CAN BE FOUND IN SECTION 5.0.

ERROR DESCRIPTIONS:

"ERROR READING SECTOR"

ERROR WAS ENCOUNTERED WHILE TRYING TO READ VERIFY THE SECTOR AFTER IT WAS WRITTEN BY THE SAME DRIVE.

"MINIMUM OF TWO DRIVES REQUIRED"

991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
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

THE PROGRAM REQUIRES AT LEAST TWO DRIVES TO PROVE COMPATABILITY.

"MAXIMUM OF FOUR DRIVES ALLOWED"

THE PROGRAM ONLY ALLOWS A MAXIMUM OF FOUR DRIVES.

"CAN'T FIND FIVE ADJACENT TRACKS"

THE PROGRAM REQUIRES TEN SETS OF FIVE ADJACENT TRACKS AT
PREDETERMINED SPOTS ACROSS THE PACK. IT WAS UNABLE TO FIND FIVE
COMPLETELY GOOD ADJACENT TRACKS IN THE LIMITS GIVEN.

"ERROR WRITING SECTOR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO WRITE THE GIVEN SECTOR.

"OVERWRITE ERROR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO READ DATA AFTER AN
OVERWRITE BY ONE DRIVE. BOTH DRIVES INVOLVED ARE GIVEN.

"READ RECOVERY ERROR"

AN ERROR WAS ENCOUNTERED WHILE TRYING TO RECOVER ANOTHER DRIVES
DATA.

"ADJACENT TRACK TEST"

AN ERROR WAS ENCOUNTERED WHILE IN THE ADJACENT TEST PART, A FURTHER
DESCRIPTION IS GIVEN.

3.2 ERROR HALTS

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

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
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

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)
 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)

1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
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

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)
BIT 0 - MUST BE ONE (1)

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

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

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)

1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
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

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

THE FOLLOWING IS A BRIEF DESCRIPTION OF THE WAY THE PROGRAM EXECUTES. THE PROGRAM WILL CHECK COMPATIBILITY BETWEEN 2 - 4 DRIVES USING THE SAME RLO1K CARTRIDGE OR SAME RLO2K CARTRIDGE. THE PROGRAM WILL ASK THE OPERATOR TO SEQUENCE THE PACK BETWEEN THE DRIVES GIVEN IN THE FOLLOWING MANNER.

PLACE PACK IN DRIVE N ON CONTROLLER X AND LOAD
UNLOAD DRIVE N ON CONTROLLER X
PLACE PACK IN DRIVE N+1 ON CONTROLLER X AND LOAD
UNLOAD DRIVE N+1 ON CONTROLLER X
ETC.....

THE PROGRAM WILL SEQUENCE IN THE ORDER THAT WAS GIVEN IN THE HARDWARE QUESTIONS. I.E.

DRIVE ? 0,1,2,3
PROGRAM WILL SEQUENCE 0,1,2,3,2,1,0
DRIVE ? 1,0,3,2
PROGRAM WILL SEQUENCE 1,0,3,2,3,0,1

1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228

WHEN THE FIRST DRIVE IS LOADED THE PROGRAM WILL ATTEMPT TO FIND TEN SETS OF FIVE ADJACENT TRACKS AT PREDETERMINED SPOTS THAT CONTAIN NO BAD SECTORS USING THE BAD SECTOR FILE. THE 10 SPOTS ARE: ON BOTH SURFACES, INNER, OUTER, MIDDLE, ONE QUARTER AND THREE QUARTERS. AFTER THIS IS DONE THE OVERWRITE TEST IS PREPARED(FIRST DRIVE CAN'T OVERWRITE) AS WELL AS THE ADJACENT TEST. AS THE PACK IS CYCLED BETWEEN DRIVES THE FOLLOWING CHECKS ARE MADE:

EACH DRIVE CAN OVERWRITE EACH OTHER DRIVE

EACH DRIVE CAN RECOVER EACH OTHERS DATA

EACH DRIVE CAN WRITE ADJACENT TO EVERY OTHER DRIVE WITHOUT DISTURBING THE OTHER'S DATA.

READS AND WRITES TAKE PLACE AFTER SEEKS FROM BOTH DIRECTIONS.

ADJACENT WRITES TAKE PLACE TO BOTH SIDES OF EACH WRITE

TESTS ARE PERFORMED AT ALL TEN SPOTS ACROSS THE PACK.

9

MACRO DEFINITIONS

```

1230      .SBTTL  MACRO DEFINITIONS
1231
1232      .MCALL  SVC
1233      002000      .=2000
1234
1235      .MACRO  WAITUS  ARG          ;MACRO MICRO-SECOND WAIT
1236      MOV    ARG,XDELAY          ;SAVE ARGUMENT
1237      JSR    PC,TIME              ;CALL TIMING ROUTING
1238      .ENDM
1239
1240      .MACRO  WAITMS  ARG          ;MACRO MILLI-SECOND WAIT
1241      MOV    ARG,YDELAY          ;SAVE ARGUMENT
1242      JSR    PC,XTIME            ;CALL TIMING ROUTINE
1243      .ENDM
1244
1245      .NLIST  CND,MD,ME
1246
1247 002000      SVC
1248      000000      SVCINS=0
1249      000000      SVCTAG=0
1250
1251      ;          POINTER NONE          ;JSD REV A
1252 002000      ;          POINTER BGNDU      ;JSD REV A
1253
1254 002000      BGNMOD  MDHEDR
1255 002000      HEADER  CNRLLA,0,0,1,PRI06      ;JSD REV A - ADDED PRI06
002000      .ASCII  /C/
002001      .ASCII  /N/
002002      .ASCII  /R/
002003      .ASCII  /L/
002004      .ASCII  /L/
002005      .BYTE   0
002006      .BYTE   0
002007      .BYTE   0
002010      .ASCII  /A/
002011      .ASCII  /O/
002012      .WORD   0
002014      .WORD   0
002016      .WORD   L$HARD
002020      .WORD   0
002022      .WORD   L$HW
002024      .WORD   0
002026      .WORD   L$LAST
002030      .WORD   0
002032      .WORD   0
002034      .WORD   1
002036      .WORD   0
002040      .WORD   L$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

```

MACRO DEFINITIONS

002062 000000
 002064 000000
 002066 000000
 002070 000000
 002072 024350
 002074 000000
 002076 002122
 002100 104035
 002102 000000
 002104 022466
 002106 024344
 002110 024340
 002112 022440
 002114 000000
 002116 000000
 002120 000000

.WORD 0
 .WORD 0
 .WORD 0
 .WORD 0
 .WORD L\$DU
 .WORD 0
 .WORD L\$DESC
 EMT E\$LOAD
 .WORD 0
 .WORD L\$INIT
 .WORD L\$CLEAN
 .WORD L\$AUTO
 .WORD L\$PROT
 .WORD 0
 .WORD 0
 .WORD 0
 ENDMOD

1256
 1257
 1258

002122
 002122 103 116 122
 002125 114 114 040
 002130 126 105 122
 002133 111 106 111
 002136 105 123 040
 002141 111 116 124
 002144 105 122 103
 002147 110 101 116
 002152 107 105 101
 002155 102 111 114
 002160 111 124 131
 002163 040 117 106
 002166 040 103 101
 002171 122 124 122
 002174 111 104 107
 002177 105 123 040
 002202 102 105 124
 002205 127 105 105
 002210 116 040 104
 002213 122 111 126
 002216 105 123 000

DESCRPT <CNRL VERIFIES INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES>
 .ASCIZ /CNRL VERIFIES INTERCHANGEABILITY OF CARTRIDGES BETWEEN DRIVES/

1259
 1260

002222
 002222 122 114 060
 002225 061 054 122
 002230 114 060 062
 002233 000

.EVEN
 DEVTYP <RL01,RL02>
 .ASCIZ %RL01,RL02%

1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268

002234
 002234

.EVEN
 .SBTTL GLOBAL EQUATES SECTION
 ;DEFINITIONS
 BGNMOD GLBEQAT
 EQUALS
 ;

C3

GLOBAL EQUATES SECTION

```

; 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
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
    
```

GLOBAL EQUATES SECTION

```

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          MOE==  100000

1269
1270          000000          CS=0          ;CONTROL AND STATUS OFFSET
1271          000002          BA=2          ;BUSS ADDRESS OFFSET
1272          000004          DA=4          ;DISK ADDRESS OFFSET
1273          000006          MP=6          ;MULTI PURPOSE OFFSET
1274
1275          ;CONSTANT OFFSETS FOR INDIVIDUAL DRIVE BUFFERS
1276
1277          000000          CSR=0          ;CONTROLLER ADDRESS
1278          000002          VEC=2          ;VECTOR OF CONTROLLER
1279          000004          DSB=4          ;DRIVE SELECT
1280          000006          PAT=6          ;PATTERN UNIQUE TO DRIVE
1281
1282          000001          DRDY=BIT0      ;DRIVE READY
1283          000100          INTEN=BIT6     ;INTERRUPT ENABLE
1284          100000          ERR=BIT15     ;COMPOSITE ERROR
1285          040000          DERR=BIT14    ;DRIVE ERROR
1286          020000          NXM=BIT13     ;NON-EXISTANT MEMORY ERROR
1287          010000          DLT=BIT12     ;DATA LATE
1288          004000          DCRC=BIT11     ;DATA CRC ERROR
1289          004000          HCRC=BIT11     ;HEADER CRC ERROR
1290          010000          HNF=BIT12     ;HEADER NOT FOUND ERROR
1291          002000          OPI=BIT10     ;OPERATION INCOMPLETE ERROR
1292          000200          CRDY=BIT7      ;CONTROLLER READY
1293          000040          BA17=BIT5     ;EXTENDED BUS ADDRESS BIT 17
1294          000020          BA16=BIT4     ;EXTENDED BUS ADDRESS BIT 16
1295          000002          CRSET=BIT1    ;CONTROLLER RESET FUNCTION CODE
1296          000004          GSTAT=BIT2    ;GET DRIVE STATUS FUNCTION CODE
1297          000006          SEEK=BIT1!BIT2 ;SEEK FUNCTION CODE
1298          000010          RDHDR=BIT3     ;READ HEADER FUNCTION CODE
1299          000012          WRITE=BIT3!BIT1 ;WRITE FUNCTION CODE
1300          000014          READ=BIT3!BIT2 ;READ FUNCTION CODE
1301          000013          DRST=BIT3!BIT1!BIT0 ;DRIVE RESET COMMAND CODE FOR DRIVE COMMAND WORD
1302          000003          GSBIT=BIT1!BIT0 ;GET STATUS COMMAND CODE FOR DRIVE COMMAND WORD
1303          000001          MK=BIT0        ;MARKER BIT FOR DRIVE COMMAND WORD(SEEK,GET STATUS)
1304          000004          SIGN=BIT2     ;DIRECTION FOR SEEK(0=AWAY FROM SPINDLE)
1305          000020          SKHS=BIT4     ;HEAD SELECT FOR SEEK
1306          000100          HEAD=BIT6     ;HEAD SELECT FOR READ,WRITE,GET STATUS
1307
1308          ;OFFSET FOR HARDWARE P-TABLE
1309
1310          000000          CSR= 0          ;BUS ADDRESS
1311          000002          VECT= 2        ;VECTOR ADDRESS
1312          000004          PRIOR= 4       ;PRIORITY (BREAK LEVEL)
1313          000006          TYPDR= 6      ;DRIVE TYPE
1314          000010          DRBT= 10      ;DRIVE SELECT

```

GLOBAL EQUATES SECTION

```

1315
1316 002234          ENDMOD
1317
1318          .SBTTL GLOBAL DATA SECTION
1319
1320 002234          BGNMOD GLBDAT
1321
1322 002234 000000    HDRFND: .WORD 0          ;1-HEADER IN BAD SECTOR LIST
1323
1324          ;HERE IS THE LIST OF TRACKS TO USE FOR THIS TEST
1325          ;TRACKS ARE ENTERED BY 'FNDTRK' ROUTINE & 'FIXTRK' ROUTINE
1326
1327 002236 000000    OUT10: .WORD 0          ;OUTER TRK HEAD 0
1328 002240 000000    OUT20: .WORD 0
1329 002242 000000    OUT30: .WORD 0
1330 002244 000000    OUT40: .WORD 0
1331 002246 000000    OUT50: .WORD 0
1332 002250 000000    OUT11: .WORD 0         ;OUTER TRK HEAD 1
1333 002252 000000    OUT21: .WORD 0
1334 002254 000000    OUT31: .WORD 0
1335 002256 000000    OUT41: .WORD 0
1336 002260 000000    OUT51: .WORD 0
1337 002262 000000    QQU10: .WORD 0        ;1ST QUARTER TRK HEAD 0
1338 002264 000000    QQU20: .WORD 0
1339 002266 000000    QQU30: .WORD 0
1340 002270 000000    QQU40: .WORD 0
1341 002272 000000    QQU50: .WORD 0
1342 002274 000000    QQU11: .WORD 0       ;1ST QUARTER TRK HEAD 1
1343 002276 000000    QQU21: .WORD 0
1344 002300 000000    QQU31: .WORD 0
1345 002302 000000    QQU41: .WORD 0
1346 002304 000000    QQU51: .WORD 0
1347 002306 000000    MID10: .WORD 0       ;MIDDLE TRK HEAD 0
1348 002310 000000    MID20: .WORD 0
1349 002312 000000    MID30: .WORD 0
1350 002314 000000    MID40: .WORD 0
1351 002316 000000    MID50: .WORD 0
1352 002320 000000    MID11: .WORD 0       ;MIDDLE TRK HEAD 1
1353 002322 000000    MID21: .WORD 0
1354 002324 000000    MID31: .WORD 0
1355 002326 000000    MID41: .WORD 0
1356 002330 000000    MID51: .WORD 0
1357 002332 000000    TQU10: .WORD 0       ;3RD QUARTER TRK HEAD 0
1358 002334 000000    TQU20: .WORD 0
1359 002336 000000    TQU30: .WORD 0
1360 002340 000000    TQU40: .WORD 0
1361 002342 000000    TQU50: .WORD 0
1362 002344 000000    TQU11: .WORD 0       ;3RD QUARTER TRK HEAD 1
1363 002346 000000    TQU21: .WORD 0
1364 002350 000000    TQU31: .WORD 0
1365 002352 000000    TQU41: .WORD 0
1366 002354 000000    TQU51: .WORD 0
1367 002356 000000    INN10: .WORD 0      ;INNER TRK HEAD 0
1368 002360 000000    INN20: .WORD 0
1369 002362 000000    INN30: .WORD 0
1370 002364 000000    INN40: .WORD 0
1371 002366 000000    INN50: .WORD 0

```

GLOBAL DATA SECTION

1372	002370	000000	INN11: .WORD	0	;INNER TRK HEAD 1
1373	002372	000000	INN21: .WORD	0	
1374	002374	000000	INN31: .WORD	0	
1375	002376	000000	INN41: .WORD	0	
1376	002400	000000	INN51: .WORD	0	
1377			.EVEN		
1378					
1379			;SECTOR LIST FOR LAST DRIVE WRITTEN		
1380			;MAP OF 16 SECTOR DRIVE BITS		
1381					
1382	002402		SECLST: .BLKW	16.	
1383					
1384			;BUFFER TABLE FOR 24 X 5 MATRIX USED FOR ADJACENT CYLINDER TESTING.		
1385					
1386	002442		SECBUF: .BLKW	5*24.	
1387					
1388			;LIST OF TRACKS USED TO OVERWRITE TEST.		
1389			;FIRST FIVE ARE CYLINDER ADDRESSES OF TOP SURFACE.		
1390			;LAST FIVE ARE CYLINDER ADDRESSES OF BOTTOM SURFACE.		
1391					
1392	003022	002242	OVWTRK: OUT30		
1393	003024	002266	OQU30		
1394	003026	002312	MID30		
1395	003030	002336	TQU30		
1396	003032	002362	INN30		
1397	003034	002254	OUT31		
1398	003036	002300	OQU31		
1399	003040	002324	MID31		
1400	003042	002350	TQU31		
1401	003044	002374	INN31		
1402					
1403	003046	152525	PATLST: .WORD	152525	
1404	003050	133333	.WORD	133333	
1405	003052	066666	.WORD	066666	
1406	003054	155555	.WORD	155555	
1407					
1408	003056	000000	TEM: .WORD	0	
1409	003060	000000	T.DRIVE: .WORD	0	
1410	003062	000000	FOUR: .WORD	0	
1411	003064	000000	FADJ: .WORD	0	
1412	003066	000000	TEMP: .WORD	0	
1413	003070	000000	LSTCLR: .WORD	0	;LAST CONTROLLER
1414	003072	000000	REASON: .WORD	0	;DRIVE ERROR REASON
1415	003074	000000	ERFLG: .WORD	0	;ERROR FLAG
1416	003076	000000	STFLG: .WORD	0	;PROGRAM START UP FLAG
1417	003100	000000	ADJLOC: .WORD	0	;TRACK INDEX FOR ADJ. CYL TEST
1418	003102	000000	ADJFLG: .WORD	0	;FLAG FOR ADJ. STORE OR RETRIEVE
1419	003104	000000	ADJDIR: .WORD	0	;ADJACENT SEEK DIRECTION
1420	003106	000000	DRSTAT: .WORD	0	
1421	003110	000000	MSFLG: .WORD	0	
1422	003112	000000	OSECT: .WORD	0	
1423	003114	000000	HEAD01: .WORD	0	;SURFACE FLAG
1424	003116	000000	DIRC: .WORD	0	;DIRECTION OF SEEK
1425	003120	000000	SURF: .WORD	0	
1426	003122	000000	CYL: .WORD	0	
1427	003124	000000	REVSK: .WORD	0	;REVERSE SEEK
1428	003126	000000	FORSK: .WORD	0	;FORWARD SEEK

GLOBAL DATA SECTION

1429	003130	000000	UUT:	.WORD	0	;UNIT UNDER TEST
1430	003132	000000	SECT:	.WORD	0	;SECTOR
1431	003134	000000	LSTDRV:	.WORD	0	;LAST DRIVE
1432	003136	000000	GDATA:	.WORD	0	;GOOD DATA
1433	003140	000000	BDATA:	.WORD	0	;BAD DATA
1434	003142	000000	WCOUNT:	.WORD	0	;WORD COUNT
1435	003144	000000	SECWRD:	.WORD	0	;SECTOR WORD
1436	003146	000000	OFFSET:	.WORD	0	;INCREMENT
1437	003150	000000	LSTTRK:	.WORD	0	;LAST TRACK OF SEARCH
1438	003152	000000	FRTTRK:	.WORD	0	;FIRST TRACK OF SEARCH
1439	003154	000000	PRSTRK:	.WORD	0	;PRESENT TRACK
1440	003156	000000	SURFACE:	.WORD	0	;SURFACE
1441	003160	000000	TRKFND:	.WORD	0	;TRACK FOUND
1442	003162	000000	TRKCNT:	.WORD	0	;TRACK COUNT
1443	003164	000000	E.CS:	.WORD	0	;IMAGE OF CSR
1444	003166	000000	E.BA:	.WORD	0	;IMAGE OF BUS ADDRESS
1445	003170	000000	E.DA:	.WORD	0	;IMAGE OF DISK ADDRESS
1446	003172	000000	E.MP:	.WORD	0	;IMAGE OF MULTI-PURPOSE WORD 1
1447	003174	000000	E.MP1:	.WORD	0	; " " " " " 2
1448	003176	000000	E.MP2:	.WORD	0	; " " " " " 3
1449	003200	000000	BCS:	.WORD	0	;COMMAND LOADED
1450	003202	000000	BBA:	.WORD	0	;BUS ADDRESS LOADED
1451	003204	000000	BDA:	.WORD	0	;DISK ADDRESS LOADED
1452	003206	000000	BMP:	.WORD	0	;WORD COUNT LOADED
1453	003210	000000	SERNM1:	.WORD	0	;SERIAL NUMBER OF CATRIDGE
1454	003212	000000	SERNM2:	.WORD	0	; " " " " "
1455	003214	000000	ADJTRK:	.WORD	0	;INSIDE/OUTSIDE FLAG
1456	003216	000000	ADJUUT:	.WORD	0	;UUT FOR "ADJCYL"
1457	003220	000000	ADJLC2:	.WORD	0	;TEMP LOC FOR "ADJCYL"
1458	003222	000000	ADJLC3:	.WORD	0	; " " " " "
1459	003224	000000	ADJLC4:	.WORD	0	; " " " " "
1460	003226	000000	STSEC1:	.WORD	0	;SECTORS TO WRITE "ADJCYL"
1461	003230	000000	STSEC:	.WORD	0	; " " " " "
1462	003232		BUF:	.BLKW	3072.	;BUFFER FOR 24 SECTOR READS
1463	017232	000000	XDELAY:	.WORD	0	;DELAY FOR WAIT MICRO-SECOND MACRO
1464	017234	000000	YDELAY:	.WORD	0	;DELAY FOR WAIT MILLI-SECOND MACRO
1465	017236	000000	OBUF:	.WORD	0	;RESPONSE BUFFER
1466						
1467	017240		DRBUF:			;DRIVE INFORMATION BUFFERS
1468						
1472						
1473		000004	.REPT	4.		
1480						
	017240	000000		CSR		;CONTROLLER ADDRESS
	017242	000002		VEC		;VECTOR
	017244	000004		DSB		;DRIVE SELECT BITS
	017246	000006		PAT		;PATTERN UNIQUE TO DRIVE
	017250	000000		CSR		;CONTROLLER ADDRESS
	017252	000002		VEC		;VECTOR
	017254	000004		DSB		;DRIVE SELECT BITS
	017256	000006		PAT		;PATTERN UNIQUE TO DRIVE
	017260	000000		CSR		;CONTROLLER ADDRESS
	017262	000002		VEC		;VECTOR

GLOBAL DATA SECTION

```

017264 000004          DSB          ;DRIVE SELECT BITS
017266 000006          PAT          ;PATTERN UNIQUE TO DRIVE

017270 000000          CSR          ;CONTROLLER ADDRESS
017272 000002          VEC          ;VECTOR
017274 000004          DSB          ;DRIVE SELECT BITS
017276 000006          PAT          ;PATTERN UNIQUE TO DRIVE

1481
1485 017300 000000          ENDBUF: .WORD 0          ;END OF DRIVE BUFFERS
1486 017302          ENDMOD
1487
1488          .SBTTL GLOBAL TEXT SECTION
1489 017302          BGNMOD GLBTXT
1490
1491          ;GLOBAL TEXT
1492
1496
1497 017302          103          117          116 OPRO01: .ASCIZ /CONTINUE TEST?/
1498 017321          101          102          117 OPRO02: .ASCIZ /ABOVE CONDITIONS MET/
1499 017346          103          117          116 CNTTOT: .ASCIZ /CONTROLLER TIMED OUT/
1500 017373          105          122          122 INITWR: .ASCIZ /ERROR ON RECOVERING INITIAL WRITE BY FIRST DRIVE /
1501 017455          105          122          122 DCKER: .ASCIZ /ERROR ON READ/
1502 017473          115          111          116 FEW: .ASCIZ /MINIMUM OF TWO DRIVES REQUIRED/
1503 017532          115          101          130 MANY: .ASCIZ /MAXIMUM OF FOUR DRIVES ALLOWED/
1504 017571          124          105          123 NONE: .ASCIZ /TEST ABORTED - CAN'T FIND ANY GOOD SPOTS/
1505 017642          124          122          131 OVMES: .ASCIZ /TRYING TO OVERWRITE DRIVE /
1506 017675          124          122          131 RECHS: .ASCIZ /TRYING TO READ DATA WRITTEN BY DRIVE /
1507 017743          103          101          116 ERRFND: .ASCIZ /CAN'T FIND FIVE ADJACENT TRACKS/
1508 020003          117          126          105 OVWER: .ASCIZ /OVERWRITE ERROR/
1509 020023          122          105          101 RECER: .ASCIZ /READ RECOVERY ERROR/
1510 020047          105          122          122 FUNERR: .ASCIZ /ERROR IN SEEK OPERATION/
1511 020077          115          111          123 SKER: .ASCIZ /MIS SEEK ERROR/
1512 020116          106          117          122 FWD: .ASCIZ /FORWARD/
1513 020126          122          105          126 REV: .ASCIZ /REVERSE/
1514 020136          105          122          122 WRIT1: .ASCIZ /ERROR WRITING SECTOR/
1515 020163          105          122          122 READ1: .ASCIZ /ERROR READING SECTOR/
1516 020210          101          104          112 ADJTXT: .ASCIZ /ADJACENT CYLINDER TEST/
1517          .EVEN
1518
1519 020240          ENDMOD
1520
1521          .SBTTL GLOBAL ERROR REPORT SECTION
1522
1523 020240          BGNMOD GLBERR
1524
1525 020240          BGNMSG ERR1
1526
1527 020240          PRINTB #FRM10,FRTRK,LSTTRK,SURFACE ;BETWEEN _ _ HEAD _
020240 013746 003156          MOV SURFACE,-(SP)
020244 013746 003150          MOV LSTTRK,-(SP)
020250 013746 003152          MOV FRTRK,-(SP)
020254 012746 021545          MOV #FRM10,-(SP)
020260 012746 000004          MOV #4,-(SP)
020264 010600          MOV SP,R0
020266 104414          TRAP C#PNTB
    
```

GLOBAL ERROR REPORT SECTION

```

1528 020270 062706 000012          ADD    #12,SP
1529 020274          ENDMSG
      020274          L10000: TRAP    C#MSG
1530 020274 104423
1531 020276          BGNMSG ERR2
1532 020276          PRINTB #FRM4,CSR(R4),<B,DSB+1(R4)> ;CONTROLLER _ DRIVE _
      020276 005046          CLR    -(SP)
      020300 156416 000005          BISB  DSB+1(R4),(SP)
      020304 016446 000000          MOV   CSR(R4),-(SP)
      020310 012746 021246          MOV   #FRM4,-(SP)
      020314 012746 000003          MOV   #3,-(SP)
      020320 010600          MOV   SP,RO
      020322 104414          TRAP  C#PNTB
      020324 062706 000010          ADD   #10,SP
1533 020330 004737 026550          JSR   PC,REGDMP ;REGISTER DUMP ROUTINE
1534 020334          ENDMSG
      020334 104423          L10001: TRAP    C#MSG
1535 020336          BGNMSG ERR3
1536 020336          PRINTB #FRM4,CSR(R4),<B,DSB+1(R4)> ;CONTROLLER _ DRIVE _
1537 020336 005046          CLR    -(SP)
      020340 156416 000005          BISB  DSB+1(R4),(SP)
      020344 016446 000000          MOV   CSR(R4),-(SP)
      020350 012746 021246          MOV   #FRM4,-(SP)
      020354 012746 000003          MOV   #3,-(SP)
      020360 010600          MOV   SP,RO
      020362 104414          TRAP  C#PNTB
      020364 062706 000010          ADD   #10,SP
1538 020370 004737 026550          JSR   PC,REGDMP ;REGISTER DUMP ROUTINE
1539 020374          PRINTB #FRM5,<SURF>,<CYL>,<SECT> ;HEAD _ CYLINDER _ SECTOR _
      020374 013746 003132          MOV   SECT,-(SP)
      020400 013746 003122          MOV   CYL,-(SP)
      020404 013746 003120          MOV   SURF,-(SP)
      020410 012746 021307          MOV   #FRM5,-(SP)
      020414 012746 000004          MOV   #4,-(SP)
      020420 010600          MOV   SP,RO
      020422 104414          TRAP  C#PNTB
      020424 062706 000012          ADD   #12,SP
1540 020430          PRINTB #FRM16,CSR(R3),<B,DSB+1(R3)> ;ADJACENT WRITTEN BY CONTROLLER
1541 020430 005046          CLR    -(SP) ;_ DRIVE _
      020432 156316 000005          BISB  DSB+1(R3),(SP)
      020434 016346 000000          MOV   CSR(R3),-(SP)
      020442 012746 022076          MOV   #FRM16,-(SP)
      020446 012746 000003          MOV   #3,-(SP)
      020452 010600          MOV   SP,RO
      020454 104414          TRAP  C#PNTB
      020456 062706 000010          ADD   #10,SP
1542 020462          ENDMSG
1543 020462          L10002: TRAP    C#MSG
      020462 104423
1544 020464          BGNMSG ERR4

```

GLOBAL ERROR REPORT SECTION

1546					
1547	020464			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)> ;CONTROLLER _ DRIVE _
	020464	005046		CLR	-(SP)
	020466	156416	000005	BISB	DSB+1(R4),(SP)
	020472	016446	000000	MOV	CSR(R4),-(SP)
	020476	012746	021246	MOV	#FRM4,-(SP)
	020502	012746	000003	MOV	#3,-(SP)
	020506	010600		MOV	SP,RO
	020510	104414		TRAP	C#PNTB
	020512	062706	000010	ADD	#10,SP
1548	020516	004737	026550	JSR	PC,REGDMP ;REGISTER DUMP ROUTINE
1549	020522			PRINTB	#FRM5,<SURF>,<CYL>,SECT ;HEAD _ CYLINDER _ SECTOR _
	020522	013746	003132	MOV	SECT,-(SP)
	020526	013746	003122	MOV	CYL,-(SP)
	020532	013746	003120	MOV	SURF,-(SP)
	020536	012746	021307	MOV	#FRM5,-(SP)
	020542	012746	000004	MOV	#4,-(SP)
	020546	010600		MOV	SP,RO
	020550	104414		TRAP	C#PNTB
	020552	062706	000012	ADD	#12,SP
1550	020556			PRINTB	#FRM6,REASON,LSTDRV,LSTCLR,LSTDRV
	020556	013746	003134	MOV	LSTDRV,-(SP)
	020562	013746	003070	MOV	LSTCLR,-(SP)
	020566	013746	003134	MOV	LSTDRV,-(SP)
	020572	013746	003072	MOV	REASON,-(SP)
	020576	012746	021356	MOV	#FRM6,-(SP)
	020602	012746	000005	MOV	#5,-(SP)
	020606	010600		MOV	SP,RO
	020610	104414		TRAP	C#PNTB
	020612	062706	000014	ADD	#14,SP
1551	020616			PRINTB	#FRM7,DIRC ;SEEK DIRECTION
	020616	013746	003116	MOV	DIRC,-(SP)
	020622	012746	021377	MOV	#FRM7,-(SP)
	020626	012746	000002	MOV	#2,-(SP)
	020632	010600		MOV	SP,RO
	020634	104414		TRAP	C#PNTB
	020636	062706	000006	ADD	#6,SP
1552					
1553	020642			ENDMSG	
	020642			L10003:	
	020642	104423		TRAP	C#MSG
1554					
1555	020644			BGNMSG	
1556	020644			PRINTB	#FRM4,CSR(R4),<B,DSB+1(R4)> ;CONTROLLER _ DRIVE _
	020644	005046		CLR	-(SP)
	020646	156416	000005	BISB	DSB+1(R4),(SP)
	020652	016446	000000	MOV	CSR(R4),-(SP)
	020656	012746	021246	MOV	#FRM4,-(SP)
	020662	012746	000003	MOV	#3,-(SP)
	020666	010600		MOV	SP,RO
	020670	104414		TRAP	C#PNTB
	020672	062706	000010	ADD	#10,SP
1557	020676	004737	026550	JSR	PC,REGDMP
1558	020702			ENDMSG	
	020702			L10004:	
	020702	104423		TRAP	C#MSG
1559					

GLOBAL ERROR REPORT SECTION

```

1560 020704
1561 020704
      020704 005046
      020706 156416 000005
      020712 016446 000000
      020716 012746 021246
      020722 012746 000003
      020726 010600
      020730 104414
      020732 062706 000010
1562 020736 004737 026550
1563 020742
      020742 013746 003172
      020746 010146
      020750 012746 022163
      020754 012746 000003
      020760 010600
      020762 104414
      020764 062706 000010
1564 020770
      020770
      020770 104423
1565
1566
1567
1571
1572 020772 045 116
1573 021067 045 116
1574 021167 045 116
1575 021246 045 101
1576 021307 045 101
1577 021356 045 124
1578 021377 045 101
1579 021437 045 101
1580 021503 045 104
1581 021545 045 101
1582 021611 045 116
1583 021646 045 101
1584 021725 045 116
1585 022010 045 116
1586 022037 045 116
1587 022076 045 101
1588 022163 045 101
1589 022217 045 116
1590 022303 045 116
1591 022341 045 116
1592 022412 045 116
1593
1597
1598
1599 022440
1600
1601
1602
1603 022440
1604
1605 022440 000000
    
```

```

BGNMSG ERR6
PRINTB @FRM4,CSR(R4),<B,DSB+1(R4)>
CLR -(SP)
BISB DSB+1(R4),(SP)
MOV CSR(R4),-(SP)
MOV @FRM4,-(SP)
MOV @3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @10,SP
JSR PC,REGDMP
PRINTB @FRM17,R1,E.MP
MOV E.MP,-(SP)
MOV R1,-(SP)
MOV @FRM17,-(SP)
MOV @3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @10,SP
ENDMSG
    
```

```

L10005: TRAP C$MSG
    
```

;FORMAT STATEMENTS

```

045 FRM1: .ASCIZ /#N#AUNLOAD DRIVE #01#A ON CONTROLLER #06#A AND REMOVE PACK#N/
045 FRM2: .ASCIZ /#N#APLACE PACK IN DRIVE #01#A ON CONTROLLER #06#A AND LOAD IT#N/
045 FRM3: .ASCIZ !#N#AWRONG PACK # IS #05#05#A # S/B #05#05#N#N!
045 FRM4: .ASCIZ /#ACONTROLLER: #06#A DRIVE: #01#N/
045 FRM5: .ASCIZ /#AHEAD: #01#A CYL: #Z3#A SECTOR: #Z2#N/
045 FRM6: .ASCIZ /#T#01#A ON #06#N/
045 FRM7: .ASCIZ /#ASEEK DIRECTION: #T#N#A#DATA:#N/
045 FRM8: .ASCIZ !#AWORD: #Z3#A S/B: #06#A WAS: #06#N!
045 FRM9: .ASCIZ /#D3#A WORDS BAD OUT OF 128 READ#N/
045 FRM10: .ASCIZ /#ABETWEEN #Z3#A - #Z3#A HEAD: #01#N/
045 FRM11: .ASCIZ /#N#APWR FAIL NOT SUPPORTED#N/
045 FRM12: .ASCIZ /#ABEFORE CS: #06#A BA: #06#A DA: #06#A MP: #06#N/
045 FRM13: .ASCIZ /#N#AAFTER CS: #06#A BA: #06#A DA: #06#A MP: #06#N/
045 FRM14: .ASCIZ /#N#A DRIVE STATUS: #06#N/
045 FRM15: .ASCIZ /#N#ACAN'T FIND BAD SECTOR FILE/
045 FRM16: .ASCIZ /#AADJACENT WRITTEN BY CONTROLLER: #06#A DRIVE: #01#N/
045 FRM17: .ASCIZ /#AEXP'D: #06#A REC'D: #06#N/
045 FRM18: .ASCIZ /#N#AUNLOAD AND WRITE ENABLE ALL DRIVES TO BE USED#N/
045 FRM19: .ASCIZ /#N#ADRIVE TYPE IS DIFFERNT.#N/
045 FRM20: .ASCIZ /#N#ADRIVE NUMBER PREVIOUSLY SPECIFIED.#N/
045 ENDPAS: .ASCIZ /#N#A END OF TEST#N#N/
    
```

.EVEN

ENDMOD

;LOAD PROTECTION TABLE

BGNPROT

.WORD 0 ;OFFSET OF CSR IN P-TABLE

GLOBAL ERROR REPORT SECTION

```

1606 022442 177777          .WORD  -1          ;NOT A MASS-BUS DRIVE
1607 022444 000006          .WORD  6          ;OFFSET OF DRIVE IN P-TABLE
1608
1609 022446                ENDPROT
1610
1611 022446                BGNMOD  HPTCODE
1612 022446                BGNHW
                                .WORD  L10007-L$HW/2
                                .WORD  174400          ;BASE ADDRESS DEFAULT
1613 022450 174400          .WORD  174400          ;VECTOR DEFAULT
1614 022452 000160          .WORD  160           ;PRIORITY DEFAULT
1615 022454 000240          .WORD  240           ;RL01 OR RL02 (RL01=1)
1616 022456 000001          .WORD  1             ;DRIVE NUMBER DEFAULT
1617 022460 000000          .WORD  0
1618 022462                ENDPHW
                                L10007:
1619
1620 022462                ENDMOD
1621
1622 022462                BGNMOD  DSPCODE
1623
1624 022462                DISPATCH 1
                                .WORD  1
                                .WORD  T1
1625
1626 022466                ENDMOD
1627
1628                .SBTTL  INITIALIZATION SECTION
1629
1630 022466                BGNMOD  INITCODE
1631
1632 022466                BGNINIT
1633
1634                :
1635 022466                :   SETPRI  #340
                                SETVEC  #140,#170000,PRI07      ;ODT ROM ADDRESS      ;JSD REV A
                                MOV     PRI07,-(SP)              ;JSD REV A
                                MOV     #170000,-(SP)
                                MOV     #140,-(SP)
                                MOV     #3,-(SP)
                                TRAP   C$SVEC
                                ADD    #10,SP
1636 022514                SETPRI  #300                          ;JSD REV A
                                MOV     #300,R0
                                TRAP   C$SPRI
1637
1638 022522 023727 002012 000002  CMP     L$UNIT,#2          ;MORE THAN TWO
1639 022530 002006                BGE     90$              ;YES, OKAY
1640
1641 022532                ERRSF  19,FEW                          ;MINIMUM OFF TWO DRIVE REQUIRED
                                TRAP   C$ERSF
                                .WORD  19
                                .WORD  FEW
                                .WORD  0
1642 022542 000137 024314                JMP     CMPENA          ;CLEAN CODE WHEN < 2 DRIVES
1643
1644 022546 023727 002012 000004 90$:  CMP     L$UNIT,#4          ;MORE THAN FOUR
1645 022554 003406                BLE     91$              ;NO, OKAY
1646

```

INITIALIZATION SECTION

```

1647 022556          ERRSF 20.,MANY          ;MAXIMUM OF FOUR DRIVES ALLOWED
      022556 104454  TRAP  C$ERSF
      022560 000024  .WORD 20
      022562 017532  .WORD MANY
      022564 000000  .WORD 0
1648 022566 000137 024314  JMP  CMPENA          ;CLEAN CODE WHEN > 4 DRIVES
1649
1650 022572 013737 002012 003130 91$:  MOV  L$UNIT,UUT          ;GET NUMBER OF UNITS
1651 022600 005001  CLR  R1                ;INIT P-TABLE
1652 022602 012704 017240  MOV  #DRBUF,R4          ;SET UP DRIVE BUFFER
1653 022606 012702 003046  MOV  #PATLST,R2        ;GET LIST OF PATTERNS
1654 022612 005737 003130 1$:  TST  UUT                ;ANY P-TABLES LEFT?
1655 022616 001513  BEQ  END                ;NO,GO TO END
1656 022620          GPHARD R1,R0          ;GET A P-TABLE
      022620 010100  MOV  R1,R0
      022622 104442  TRAP  C$GPHRD
1657 022624 012064 000000  MOV  (R0)+,CSR(R4)      ;GET CSR
1658 022630 012064 000002  MOV  (R0)+,VEC(R4)      ;GET VECTOR
1659 022634 012064 000004  MOV  (R0)+,PRIOR(R4)   ;GET BREAK LEVEL
1660 022640 012037 003060  MOV  (R0)+,T.DRIVE     ;RL01/2 TYPE ... RL01=1
1661 022644 011064 000004  MOV  (R0),DSB(R4)      ;GET DRIVE
1662 022650 011264 000006  MOV  (R2),PAT(R4)
1663 022654 005722  TST  (R2)+
1664
1665          ;TEST FOR DRIVES OF SAME TYPE AND NO REPEATED DRIVE NUMBERS
1666
1667 022656 023737 002012 003130  CMP  L$UNIT,UUT          ;SKIP TEST FOR FIRST DRIVE
1668 022664 001462  BEQ  6$
1669
1670 022666          GPHARD #0,R5          ;BASE ADDRESS OF FIRST P TABLE
      022666 012700 000000  MOV  #0,R0
      022672 104442  TRAP  C$GPHRD
      022674 010005  MOV  R0,R5
1671
1672 022676 023765 003060 000006  CMP  T.DRIVE,TYPDR(R5) ;CHECK DRIVE TYPE
1673 022704 001423  BEQ  4$
1674 022706          PRINTF #FRM19          ;PROMPT - DRIVE TYPE DIFFERNT ...
      022706 012746 022303  MOV  #FRM19,-(SP)
      022712 012746 000001  MOV  #1,-(SP)
      022716 010600  MOV  SP,R0
      022720 104417  TRAP  C$PNTF
      022722 062706 000004  ADD  #4,SP
1675
1676 022726          GMANIL OPRO01,OBUFF,1,YES ;PROMPT - CONTINUE TEST
      022726 104443  TRAP  C$GMAN
      022730 000404  BR  10000$
      022732 017236  .WORD  OBUFF
      022734 000130  .WORD  T$CODE
      022736 017302  .WORD  OPRO01
      022740 000001  .WORD  1
      022742          10000$:
1677 022742 005737 017236  TST  OBUFF
1678 022746 001002  BNE  4$
1679 022750 000137 024314  JMP  CMPENA          ;RETURN TO SUPERVISOR
1680
1681 022754 026465 000004 000010 4$:  CMP  DSB(R4),DRBT(R5) ;CHECK DRIVE NUMBER
1682 022762 001023  BNE  6$

```

INITIALIZATION SECTION

```

1683 022764          PRINTF  #FRM20          ;PROMPT - DRIVE NUMBER ...
      022764 012746 022341  MOV      #FRM20,-(SP)
      022770 012746 000001  MOV      #1,-(SP)
      022774 010600          MOV      SP,R0
      022776 104417          TRAP     C$PNTF
      023000 062706 000004  ADD      #4,SP

1684
1685 023004          GMANIL  OPROO1,OBUFF,1,YES ;PROMPT - CONTINUE TEST
      023004 104443          TRAP     C$GMAN
      023006 000404          BR       10001$
      023010 017236          .WORD   OBUFF
      023012 000130          .WORD   T$CODE
      023014 017302          .WORD   OPROO1
      023016 000001          .WORD   1
      023020          10001$:
1686 023020 005737 017236  TST      OBUFF
1687 023024 001002          BNE     6$
1688 023026 000137 024314  JMP      CMPENA          ;RETURN TO SUPERVISOR
1689
1690 023032 005201          6$:   INC      R1          ;NEXT P TABLE
1691 023034 005337 003130  DEC      UUT          ;NEXT DRIVE
1692 023040 062704 000010  ADD      #PAT+2,R4
1693 023044 000662          BR       1$
1694 023046 013737 002012 003130  END:   MOV      L$UNIT,UUT
1695 023054 012704 017240          MOV      #DRBUF,R4          ;GET BEGINNING OF BUFFER
1696 023060 005037 003064          CLR      FADJ          ;CLEAR ADJ. TEST FLAG
1697 023064 005037 003062          CLR      FOWR          ;CLEAR OVERWRITE FLAG
1698 023070          READEF  #EF.PWR
      023070 012700 000034  MOV      #EF.PWR,R0
      023074 104447          TRAP     C$REFG
1699 023076          BNCOMPLETE  SETUP
      023076 103010          BCC     SETUP
1700 023100          PRINTF  #FRM11          ;PROMPT - PWR FAIL NOT SUPPORTED
      023100 012746 021611  MOV      #FRM11,-(SP)
      023104 012746 000001  MOV      #1,-(SP)
      023110 010600          MOV      SP,R0
      023112 104417          TRAP     C$PNTF
      023114 062706 000004  ADD      #4,SP

1701
1702          ;INITIALIZE ROUTINE
1703          ;WE ATTEMPT TO LOCATE 5 PERFECT ADJACENT TRACKS AT 5 SPOTS
1704          ;ACROSS THE PACK.
1705          ;THE 5 SPOTS ARE: (EACH SURFACE)
1706          ;
1707          ;OUTER - TRACK 0 - 16 (BOTH RL01 & RL02)
1708          ;INNER - TRACK 238 - 254 (RL01) OR 494 - 510 (RL02)
1709          ;MIDDLE - TRACK 120 - 136 (RL01) OR 248 - 264 (RL02)
1710          ;ONE QUARTER - TRACK 56 - 72 (RL01) OR 120 - 136 (RL02)
1711          ;THREE QUARTER - TRACK 184 - 200 (RL01) OR 376 - 392 (RL02)
1712          ;
1713          ;IF WE FIND ANY BAD SPOTS, WE WILL REPORT SO.....
1714
1715 023120 005237 003076  SETUP:  INC      STFLG          ;INDICATE A START COMMAND
1716 023124 012737 177777 003210  MOV      #-1,SERNM1
1717 023132 012737 177777 003212  MOV      #-1,SERNM2
1718 023140          1$:   PRINTF  #FRM18          ;PROMPT - UNLOAD DRIVES TO BE USED
      023140 012746 022217  MOV      #FRM18,-(SP)

```


INITIALIZATION SECTION

```

023144 012746 000001      MOV      #1, -(SP)
023150 010600      MOV      SP, RO
023152 104417      TRAP     C$PNTF
023154 062706 000004      ADD      #4, SP
1719 023160      GMANIL   OPR002, OBUFF, 1, NO      ;PROMPT - ABOVE CONDITIONS MET
023160 104443      TRAP     C$GMAN
023162 000404      BR       10002$
023164 017236      .WORD   OBUFF
023166 000120      .WORD   T$CODE
023170 017321      .WORD   OPR002
023172 000001      .WORD   1
023174      10002$:
1720 023174 005737 017236      TST      OBUFF                      ;NO - ASK AGAIN
1721 023200 001757      BEQ     1$
1722
1723 023202 004537 032326      JSR      R5, LOAD                    ; TELL OPERATOR TO LOAD
1724 023206 004537 031554      JSR      R5, SERNUM                  ; GET SERIAL NUMBER
1725 023212 004537 031030      JSR      R5, MERGE                    ; MERGE BAD SECTOR FILES
1726 023216 012701 002236      MOV      #OUT10, R1                  ; INITIALIZE ALL TRACKS
1727 023222 012700 000062      MOV      #50, RO
1728 023226 012721 177777      3$:  MOV      #177777, (R1)
1729 023232 005300      DEC     RO
1730 023234 001374      BNE     3$
1731
1732 023236 004537 031256      JSR      R5, FNDTRK                    ; TRY TO FIND FIVE TRACKS
1733 023242 000001      1$
1734 023244 000000      0$
1735
1736 023246 000000 000020      .WORD   0, 16.
1737 023252 000000 000020      .WORD   0, 16.
1738
1739 023256 005737 003160      TST      TRKFND                      ; WAS SEARCH SUCCESSFUL????
1740 023262 001005      BNE     5$                            ; YES
1741
1742 023264      ERRHRD  10, ERRFND, ERR1              ; CAN'T FIND 5 ADJACENT TRACKS
023264 104456      TRAP     C$ERRHRD
023266 000012      .WORD   10
023270 017743      .WORD   ERRFND
023272 020240      .WORD   ERR1
1743 023274 000404      BR       7$
1744
1745 023276 012700 002236      5$:  MOV      #OUT10, RO                    ; STORE AWAY TRACKS FOUND
1746 023302 004537 031520      JSR      R5, FIXCYL
1747
1748 023306 004537 031256      7$:  JSR      R5, FNDTRK                    ; TRY TO FIND FIVE TRACKS
1749 023312 000001      1$
1750 023314 000001      1$
1751 023316 000000 000020      .WORD   0, 16.
1752 023322 000000 000020      .WORD   0, 16.
1753
1754 023326 005737 003160      TST      TRKFND                      ; WAS SEARCH SUCCESSFUL????
1755 023332 001005      BNE     9$                            ; YES
1756
1757 023334      ERRHRD  10, ERRFND, ERR1              ; CAN'T FIND 5 ADJACENT TRACKS
023334 104456      TRAP     C$ERRHRD
023336 000012      .WORD   10
023340 017743      .WORD   ERRFND
    
```

INITIALIZATION SECTION

1758	023342	020240		.WORD	ERR1	
	023344	000404		BR	10#	
1759						
1760	023346	012700	002250	9#:	MOV	@OUT11,R0 ;STORE TRACKS AWAY
1761	023352	004537	031520		JSR	R5,FIXCYL
1762	023356	004537	031256	10#:	JSR	R5,FNDTRK ;FIND NEXT 5 TRACK
1763	023362	177777			-1	;OUTWARD SEARCH
1764	023364	000000			0	;TOP SURFACE
1765	023366	000376	000356		.WORD	254.,238.
1766	023372	000776	000756		.WORD	510.,494.
1767						
1768	023376	005737	003160		TST	TRKFND ;WAS SEARCH SUCCESSFUL?
1769	023402	001005			BNE	12# ;YES
1770						
1771	023404				ERRHRD	10.,ERRFND,ERR1 ;CAN'T FIND 5 ADJACENT TRACKS
	023404	104456			TRAP	C#ERRHRD
	023406	000012			.WORD	10
	023410	017743			.WORD	ERRFND
	023412	020240			.WORD	ERR1
1772	023414	000404			BR	14# ;SKIP
1773						
1774	023416	012700	002356	12#:	MOV	@INN10,R0 ;STORE AWAY TRACKS FOUND
1775	023422	004537	031520		JSR	R5,FIXCYL
1776						
1777	023426	004537	031256	14#:	JSR	R5,FNDTRK ;NEXT SET
1778	023432	177777			-1	;OUTWARD SEARCH
1779	023434	000001			1	;BOTTOM SURFACE
1780	023436	000376	000356		.WORD	254.,238.
1781	023442	000776	000756		.WORD	510.,494.
1782						
1783	023446	005737	003160		TST	TRKFND ;SEARCH SUCCESSFUL?
1784	023452	001005			BNE	16# ;YES
1785						
1786	023454				ERRHRD	10.,ERRFND,ERR1 ;CAN'T FIND 5 ADJACENT TRACKS
	023454	104456			TRAP	C#ERRHRD
	023456	000012			.WORD	10
	023460	017743			.WORD	ERRFND
	023462	020240			.WORD	ERR1
1787	023464	000404			BR	18#
1788						
1789	023466	012700	002370	16#:	MOV	@INN11,R0 ;STORE AWAY TRACKS FOUND
1790	023472	004537	031520		JSR	R5,FIXCYL
1791						
1792	023476	004537	031256	18#:	JSR	R5,FNDTRK ;NEXT SET
1793	023502	000001			1	;INWARD SEARCH
1794	023504	000000			0	;TOP SURFACE
1795	023506	000176	000210		.WORD	126.,136.
1796	023512	000376	000410		.WORD	254.,264.
1797						
1798	023516	005737	003160		TST	TRKFND ;DID WE FIND A SET
1799	023522	001020			BNE	20# ;YES
1800						
1801	023524	004537	031256		JSR	R5,FNDTRK ;NEXT SET (OTHER SIDE)
1802	023530	177777			-1	;OUTWARD SEARCH
1803	023532	000000			0	;TOP SURFACE
1804	023534	000202	000170		.WORD	130.,120.
1805	023540	000402	000370		.WORD	258.,248.

INITIALIZATION SECTION

1806	023544	005737	003160		TST	TRKFND		;DID WE FIND A SET
1807	023550	001005			BNE	20#		;YES
1808								
1809	023552				ERRHRD	10.,ERRFND,ERR1		;CAN'T FIND 5 ADJACENT TRACKS
	023552	104456			TRAP	C#ERHRD		
	023554	000012			.WORD	10		
	023556	017743			.WORD	ERRFND		
	023560	020240			.WORD	ERR1		
1810	023562	000404			BR	22#		
1811								
1812	023564	012700	002306	20#:	MOV	#MID10,R0		;STORE AWAY
1813	023570	004537	031520		JSR	R5,FIXCYL		
1814	023574	004537	031256	22#:	JSR	R5,FNDTRK		;NEXT SET
1815	023600	000001			1			;INWARD SEARCH
1816	023602	000001			1			;BOTTOM SURFACE
1817	023604	000176	000210		.WORD	126.,136.		;RANGE
1818	023610	000376	000410		.WORD	254.,264.		
1819								
1820	023614	005737	003160		TST	TRKFND		;SUCCESS?
1821	023620	001020			BNE	24#		;YES
1822								
1823	023622	004537	031256		JSR	R5,FNDTRK		;LOOK THE OTHER SIDE
1824	023626	177777			-1			;OUTWARD
1825	023630	000001			1			;BOTTOM SURFACE
1826	023632	000202	000170		.WORD	130.,120.		
1827	023636	000402	000370		.WORD	258.,248.		
1828								
1829	023642	005737	003160		TST	TRKFND		;SUCCESS?
1830	023646	001005			BNE	24#		;YES
1831								
1832	023650				ERRHRD	10.,ERRFND,ERR1		;CAN'T FIND 5 ADJACENT TRACKS
	023650	104456			TRAP	C#ERHRD		
	023652	000012			.WORD	10		
	023654	017743			.WORD	ERRFND		
	023656	020240			.WORD	ERR1		
1833	023660	000404			BR	26#		
1834								
1835	023662	012700	002320	24#:	MOV	#MID11,R0		;STORE AWAY THE TRACKS FOUND
1836	023666	004537	031520		JSR	R5,FIXCYL		
1837								
1838	023672	004537	031256	26#:	JSR	R5,FNDTRK		;NEXT SET
1839	023676	000001			1			;INWARD
1840	023700	000000			0			;TOP SURFACE
1841	023702	000076	000110		.WORD	62.,72.		;RANGE
1842	023706	000176	000210		.WORD	126.,136.		
1843								
1844	023712	005737	003160		TST	TRKFND		;SUCCESS?
1845	023716	001020			BNE	28#		;YES
1846								
1847	023720	004537	031256		JSR	R5,FNDTRK		;LOOK OTHER SIDE
1848	023724	177777			-1			;OUTWARD
1849	023726	000000			0			;TOP SURFACE
1850	023730	000102	000070		.WORD	66.,56.		;RANGE
1851	023734	000202	000170		.WORD	130.,120.		
1852								
1853	023740	005737	003160		TST	TRKFND		;SUCCESS?
1854	023744	001005			BNE	28#		;YES

INITIALIZATION SECTION

```

1855
1856 023746          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS
      023746 104456  TRAP      C#ERRHRD
      023750 000012  .WORD      10
      023752 017743  .WORD      ERRFND
      023754 020240  .WORD      ERR1
1857 023756 000404  BR          30#
1858
1859 023760 012700 002262 28#: MOV      #0QU10,R0      ;STORE AWAY NEXT SET
1860 023764 004537 031520 JSR      R5,FXCYL
1861 023770 004537 031256 30#: JSR      R5,FNDTRK      ;LOOK FOR NEXT SET
1862 023774 000001 1          ;INWARD
1863 023776 000001 1          ;BOTTOM
1864 024000 000076 000110 .WORD    62.,72.      ;RANGE
1865 024004 000176 000210 .WORD    126.,136.
1866
1867 024010 005737 003160 TST      TRKFND      ;SUCCESS?
1868 024014 001020 BNE      32#         ;YES
1869
1870 024016 004537 031256 JSR      R5,FNDTRK      ;LOOK FOR ANOTHER SET
1871 024022 177777 -1       ;OUTWARD
1872 024024 000001 1          ;BOTTOM
1873 024026 000102 000070 .WORD    66.,56.      ;RANGE
1874 024032 000202 000170 .WORD    130.,120.
1875
1876 024036 005737 003160 TST      TRKFND      ;SUCCESS?
1877 024042 001005 BNE      32#         ;YES
1878
1879 024044          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS
      024044 104456  TRAP      C#ERRHRD
      024046 000012  .WORD      10
      024050 017743  .WORD      ERRFND
      024052 020240  .WORD      ERR1
1880 024054 000404  BR          34#
1881
1882 024056 012700 002274 32#: MOV      #0QU11,R0      ;STORE AWAY TRACKS
1883 024062 004537 031520 JSR      R5,FXCYL
1884
1885 024066 004537 031256 34#: JSR      R5,FNDTRK      ;NEXT SET OF TRACKS
1886 024072 000001 1          ;INWARD
1887 024074 000000 0          ;TOP SURFACE
1888 024076 000276 000310 .WORD    190.,200.    ;RANGE
1889 024102 000576 000610 .WORD    382.,392.
1890
1891 024106 005737 003160 TST      TRKFND      ;SUCCESS?
1892 024112 001020 BNE      36#         ;YES
1893
1894 024114 004537 031256 JSR      R5,FNDTRK      ;LOOK OTHER SIDE
1895 024120 177777 -1       ;OUTWARD SEARCH
1896 024122 000000 0          ;TOP
1897 024124 000302 000270 .WORD    194.,184.
1898 024130 000602 000570 .WORD    386.,376.
1899
1900 024134 005737 003160 TST      TRKFND      ;SUCCESS
1901 024140 001005 BNE      36#         ;YES
1902
1903 024142          ERRHRD 10.,ERRFND,ERR1      ;CAN'T FIND 5 ADJACENT TRACKS

```

INITIALIZATION SECTION

```

024142 104456 TRAP C$ERHRD
024144 000012 .WORD 10
024146 017743 .WORD ERRFND
024150 020240 .WORD ERR1
1904 024152 000404 BR 38$
1905
1906 024154 012700 002332 36$: MOV #TQU10,R0 ;STORE TRACKS AWAY
1907 024160 004537 031520 JSR R5,FIXCYL
1908 024164 004537 031256 38$: JSR R5,FNDTRK ;NEXT SET
1909 024170 000001 1 ;INWARD
1910 024172 000001 1 ;BOTTOM SURFACE
1911 024174 000276 000310 .WORD 190.,200. ;RANGE
1912 024200 000576 000610 .WORD 382.,392.
1913
1914 024204 005737 003160 TST TRKFND ;SUCCESS?
1915 024210 001020 BNE 40$ ;YES
1916
1917 024212 004537 031256 JSR R5,FNDTRK ;OTHER SET
1918 024216 177777 -1 ;OUTWARD
1919 024220 000001 1 ;BOTTOM SURFACE
1920 024222 000302 000270 .WORD 194.,184. ;RANGE
1921 024226 000602 000570 .WORD 386.,376.
1922
1923 024232 005737 003160 TST TRKFND ;SUCCESS
1924 024236 001005 BNE 40$ ;YES
1925
1926 024240 ERRHRD 10.,ERRFND,ERR1 ;CZN'T FIND 5 ADJACENT TRACKS
024240 104456 TRAP C$ERHRD
024242 000012 .WORD 10
024244 017743 .WORD ERRFND
024246 020240 .WORD ERR1
1927 024250 000404 BR 42$
1928
1929 024252 012700 002344 40$: MOV #TQU11,R0 ;STORE SET AWAY
1930 024256 004537 031520 JSR R5,FIXCYL
1931
1932 024262 012700 002236 42$: MOV #OUT10,R0 ;DID WE FIND ANY AT ALL
1933 024266 012701 000062 MOV #50.,R1
1934 024272 022720 177777 44$: CMP #-1,(R0)+
1935 024276 001017 BNE EXIT
1936 024300 005301 DEC R1
1937 024302 001373 BNE 44$
1938 024304 ERRSF 3.,NONE
024304 104454 TRAP C$ERSF
024306 000003 .WORD 3
024310 017571 .WORD NONE
024312 000000 .WORD 0
1939 024314 005001 CMPENA: CLR R1
1940 ; MOV L$UNIT,R0 ;USE R2 BECAUSE R0 IS USED IN ;JSD REV A
1941 024316 013702 002012 ; MOV L$UNIT,R2 ;...DODU MACRO EXPANSION ;JSD REV A
1942 024322 48$: DODU R1 ;DO DROP UNIT
024322 010100 MOV R1,R0
024324 104451 TRAP C$DODU
1943 024326 005201 INC R1
1944 ; DEC R0 ;JSD REV A
1945 024330 005302 DEC R2 ;JSD REV A
1946 024332 001373 BNE 48$
    
```

INITIALIZATION SECTION

```

1947 024334          DOCLN
      024334 104444  TRAP      C$DCLN
1948
1949 024336          EXIT:
1950 024336          L10010:  ENDINIT
      024336 104411  TRAP      C$INIT
1951 024340          ENDMOD
1952
1953 024340          BGNMOD  AUTOCODE          ;AUTO DROP SECTION
1954 024340          BGNAUTO
1955
1956 024340 000240   NOP                      ;DO NOTHING
1957
1958 024342          ENDAUTO
      024342          L10011:
      024342 104461  TRAP      C$AUTO
1959 024344          ENDMOD
1960
1961 024344          BGNMOD  CLNCODE
1962 024344          BGNCLN
1963
1964 024344 000240   NOP
1965
1966 024346          L10012:  ENDCLN
      024346          TRAP      C$CLEAN
      024346 104412  ENDMOD
1967 024350          BGNMOD  DRPCODE
1968
1969 024350          BGNDU
1970 024350          NOP
1971 024350 000240   ENDDU
1972 024352          L10013:  TRAP      C$DU
      024352 104453  ENDMOD
1973 024354          .SBTTL  GLOBAL SUBROUTINES SECTION
1974
1975
1976
1977 024354          BGNMOD  GLBSUB
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987 024354 012737 000160 002116 TIME:  MOV     #160,L#DLY          ;GET OUTER DELAY LOOP
1988 024362 005437 017232          NEG     XDELAY          ;GET NEGATIVE OF MULTIPLY FACTOR
1989 024366          READBUS          ;Q-BUS?
      024366 104407  TRAP      C$RDBU
1990 024370          BCOMPLETE          2#          ;BRANCH - IF YES
      024370 103420  BCS     2#
1991 024372          1#:  DELAY     1          ;WAIT
      024372 012727 000001  MOV     #1,(PC)+

```

GLOBAL SUBROUTINES SECTION

	024376	000000			.WORD	0		
	024400	013727	002116		MOV	L\$DLY,(PC)+		
	024404	000000			.WORD	0		
	024406	005367	177772		DEC	-6(PC)		
	024412	001375			BNE	.-4		
	024414	005367	177756		DEC	-22(PC)		
	024420	001367			BNE	.-20		
1992	024422	005237	017232		INC	XDELAY		;WAIT FACTOR EXPIRED?
1993	024426	002761			BLT	1\$;BRANCH - IF NO
1994	024430	000422			BR	4\$;EXIT
1995	024432	012737	000150	002116	2\$: MOV	#150,L\$DLY		;GET OUTER DELAY LOOP
1996	024440				3\$: DELAY	1		;WAIT WITH RESPECT TO FONZ BUS
	024440	012727	000001		MOV	#1,(PC)+		
	024444	000000			.WORD	0		
	024446	013727	002116		MOV	L\$DLY,(PC)+		
	024452	000000			.WORD	0		
	024454	005367	177772		DEC	-6(PC)		
	024460	001375			BNE	.-4		
	024462	005367	177756		DEC	-22(PC)		
	024466	001367			BNE	.-20		
1997	024470	005237	017232		INC	XDELAY		;WAIT FACTOR EXPIRED?
1998	024474	002761			BLT	3\$;BRANCH - IF NO
1999	024476	000207			4\$: RTS	PC		;RETURN
2000								
2001	024500	012737	000160	002116	XTIME: MOV	#160,L\$DLY		;GET OUTER DELAY LOOP
2002	024506	006337	017234		ASL	YDELAY		;MULTIPLY FACTOR BY 4
2003	024512	006337	017234		ASL	YDELAY		;-----
2004	024516	005437	017234		NEG	YDELAY		;GET NEGATIVE OF RESULT
2005	024522				READBUS			;Q-BUS?
	024522	104407			TRAP	C#RDBU		
2006	024524				BNCOMPLETE	1\$;BRANCH - IF NO
	024524	103023			BCC	1\$		
2007	024526	012737	000150	002116	MOV	#150,L\$DLY		;GET OUTER DELAY LOOP
2008	024534				2\$: DELAY	20		;WAIT WITH RESPECT TO FONZ BUS
	024534	012727	000020		MOV	#20,(PC)+		
	024540	000000			.WORD	0		
	024542	013727	002116		MOV	L\$DLY,(PC)+		
	024546	000000			.WORD	0		
	024550	005367	177772		DEC	-6(PC)		
	024554	001375			BNE	.-4		
	024556	005367	177756		DEC	-22(PC)		
	024562	001367			BNE	.-20		
2009	024564	005237	017234		INC	YDELAY		;WAIT FACTOR EXPIRED?
2010	024570	002761			BLT	2\$;BRANCH - IF NO
2011	024572	000417			BR	3\$;EXIT
2012	024574				1\$: DELAY	50		;WAIT
	024574	012727	000050		MOV	#50,(PC)+		
	024600	000000			.WORD	0		
	024602	013727	002116		MOV	L\$DLY,(PC)+		
	024606	000000			.WORD	0		
	024610	005367	177772		DEC	-6(PC)		
	024614	001375			BNE	.-4		
	024616	005367	177756		DEC	-22(PC)		
	024622	001367			BNE	.-20		
2013	024624	005237	017234		INC	YDELAY		;WAIT FACTOR EXPIRED?
2014	024630	002761			BLT	1\$;BRANCH - IF NO
2015	024632	000207			3\$: RTS	PC		;RETURN

GLOBAL SUBROUTINES SECTION

```

2016
2017
2018 ;ROUTINE TO PERFORM OVERWRITE
2019 ;CALL: JSR R5,OVWPER
2020 ; SECTORS TO WRITE FORWARD
2021 ; SECTORS TO WRITE REVERSE
2022 024634 010046 OVWPER: MOV R0,-(SP) ;SAVE R0, R1, R2, R3
2023 024636 010146 MOV R1,-(SP)
2024 024640 010246 MOV R2,-(SP)
2025 024642 010346 MOV R3,-(SP)
2026 024644 005000 CLR R0 ;R0 HAS COUNT IF R0<5.
2027 024646 012537 003126 MOV (R5)+,FORSK ;USE TOP SURFACE, IF R0>5.
2028 024652 012537 003124 MOV (R5)+,REVSK ;USE BOTTOM SURFACE, IF R0>1
2029 ;DONE.
2030 024656 012701 003022 1$: MOV #OVWTRK,R1 ;GET START OF LIST OF TRACKS
2031 024662 011102 MOV (R1),R2 ;GET POINTER TO TRACK
2032 024664 021227 177777 CMP (R2),#-1 ;LEGIT TRACK??????
2033 024670 001500 BEQ 3$ ;NO, EXIT
2034
2035 024672 005037 003122 CLR CYL ;CLEAR CYLINDER/HEAD FOR SEEK
2036 024676 005037 003120 CLR SURF
2037 024702 020027 000005 CMP R0,#5 ;TOP/BOTTOM
2038 024706 002402 BLT 2$ ;TOP, BRANCH
2039 024710 005237 003120 INC SURF ;BOTTOM SURFACE
2040 024714 004537 026302 2$: JSR R5,SKCYL ;SEEK TO CYLINDER
2041 024720 005037 003122 CLR CYL
2042 024724 051237 003122 BIS (R2),CYL
2043 024730 004537 026302 JSR R5,SKCYL ;SEEK TO PROPER CYLINDER
2044 024734 013703 003126 MOV FORSK,R3 ;SECTORS TO WRITE
2045 024740 004537 025116 JSR R5,WRSEC ;GO WRITE SECTORS
2046 024744 000034 .WORD 28.
2047 024746 012737 020116 003116 MOV #FWD,DIRC ;SET FORWARD DIRECTION
2048 024754 004537 027226 JSR R5,VEROW ;VERIFY OVERWRITE
2049 024760 004537 027612 JSR R5,VEROD ;VERIFY OTHER DRIVES DATA
2050 024764 005037 003122 CLR CYL
2051 024770 022737 000001 003060 CMP #1,T.DRIVE ;RL01?
2052 024776 001004 BNE 50$ ;NO
2053 025000 052737 000377 003122 BIS #377,CYL ;SET TO GO TO MAX CYL
2054 025006 000403 BR 51$
2055 025010 052737 000777 003122 50$: BIS #777,CYL ;MAX CYL FOR RL02
2056 025016 004537 026302 51$: JSR R5,SKCYL ;SEEK TO MAX CYLINDER ON DRIVE
2057 025022 005037 003122 CLR CYL
2058 025026 005037 003120 CLR SURF
2059 025032 051237 003122 BIS (R2),CYL
2060 025036 004537 026302 JSR R5,SKCYL ;DO ANOTHER SEEK
2061
2062 025042 013703 003124 MOV REVSK,R3 ;SECTORS TO WRITE
2063 025046 004537 025116 JSR R5,WRSEC ;WRITE THEM
2064 025052 000034 .WORD 28.
2065 025054 012737 020126 003116 MOV #REV,DIRC ;SET DIRECTION
2066 025062 004537 027226 JSR R5,VEROW ;VERIFY OVERWRITE
2067 025066 004537 027612 JSR R5,VEROD ;VERIFY OTHER DRIVES DATA
2068
2069 025072 005721 3$: TST (R1)+ ;INCREMENT TO NEXT TRACK
2070 025074 005200 INC R0 ;ACCOUNT FOR IT
2071 025076 020027 000012 CMP R0,#10. ;DONE?
2072 025102 001267 BNE 1$ ;NO, GO BACK

```


GLOBAL SUBROUTINES SECTION

```

2073
2074 025104 012603      MOV      (SP)+,R3      ;RESTORE REG.
2075 025106 012602      MOV      (SP)+,R2
2076 025110 012601      MOV      (SP)+,R1
2077 025112 012600      MOV      (SP)+,R0
2078 025114 000205      RTS       R5           ;EXIT
2079
2080      ;ROUTINE TO WRITE SECTORS
2081      ;USED IN OVERWRITE TEST;ADJACENT CYLINDER TEST
2082      ;CALL JSR R5,WRSEC
2083      ;      .WRD           ;STARTING SECTOR
2084      ;R3 HAS BITMAP OF SECTORS TO WRITE
2085      ;R4 HAS DRIVE BUFFER POINTER
2086
2087 025116 010046      WRSEC:  MOV      R0,-(SP)      ;SAVE R0
2088 025120 010146      MOV      R1,-(SP)      ;SAVE R1
2089 025122 010246      MOV      R2,-(SP)      ;SAVE R2
2090 025124 012701 003232  MOV      @BUF,R1        ;WRITE PATTERN INTO
2091 025130 012702 000200  MOV      @128.,R2       ;MEMORY THAT WE
2092 025134 016421 000006  2$:  MOV      PAT(R4),(R1)+   ;WILL WRITE ONTO
2093 025140 005302      DEC      R2            ;PACK FOR THIS
2094 025142 001374      BNE     2$            ;DRIVE
2095 025144 012701 100000  MOV      @100000,R1     ;MASK FOR BIT MAP
2096 025150 012737 000007 003056  MOV      @7,TEM
2097 025156 053702 003122  BIS     CYL,R2
2098 025162 006302      120$: ASL     R2
2099 025164 005337 003056  DEC     TEM
2100 025170 001374      BNE     120$
2101 025172 005737 003120  TST     SURF
2102 025176 001402      BEQ     3$            ;0, SKIP
2103 025200 052702 000100  BIS     @HEAD,R2       ;SET BOTTOM HEAD
2104 025204 052502      3$:  BIS     (R5)+,R2     ;START AT SECTOR 28.
2105 025206 030103      4$:  BIT     R1,R3        ;WRITE THIS SECTOR?
2106 025210 001452      BEQ     5$            ;NO
2107
2108 025212 005037 003110  CLR     HSFLG
2109 025216 012737 177600 003206  MOV     @-128.,BMP     ;LOAD WORD COUNT
2110 025224 010237 003204  MOV     R2,BDA         ;LOAD DISK ADDRESS
2111 025230 010237 003066  MOV     R2,TEMP        ;SAVE DISK ADDRESS
2112 025234 042702 177700  BIC     @177700,R2
2113 025240 020227 000047  CMP     R2,@39.
2114 025244 003403      BLE     6$
2115 025246 162737 000050 003204  SUB     @40.,BDA
2116 025254 012737 003232 003202  6$:  MOV     @BUF,BBA      ;LOAD BUS ADDRESS
2117 025262 013702 003066  MOV     TEMP,R2        ;RESTORE DISK ADDRESS
2118 025266 004537 032432  11$:  JSR     R5,LDFUNC     ;GO WRITE
2119 025272 000012      WRITE
2120 025274 005737 003074  TST     ERFLG         ;ERROR IN WRITING
2121 025300 001416      BEQ     5$            ;NO,OKAY
2122 025302 005737 003110  TST     HSFLG
2123 025306 001007      BNE     10$
2124 025310      ERRSOFT 100.,WRIT1,ERR2
      025310 104457      TRAP   C#ERSOFT
      025312 000144      .WORD 100
      025314 020136      .WORD WRIT1
      025316 020276      .WORD ERR2
2125 025320 005237 003110  INC     HSFLG

```

GLOBAL SUBROUTINES SECTION

```

2126 025324 000760
2127 025326 104456 10$: BR 11$
      025326 104456      ERRHRD 110.,WRIT1,ERR2
      025330 000156      TRAP C$ERHRD
      025332 020136      .WORD 110
      025334 020276      .WORD WRIT1
                                .WORD ERR2

2128
2129 025336 005202 5$: INC R2 ;NEXT SECTOR
2130 025340 000241      CLC ;CLEAR CARRY BIT
2131 025342 006001      ROR R1 ;DONE?
2132 025344 103320      BCC 4$ ;NO GO BACK
2133 025346 012602      MOV (SP)+,R2 ;RESTORE REGISTERS AND EXIT
2134 025350 012601      MOV (SP)+,R1
2135 025352 012600      MOV (SP)+,R0
2136 025354 000205      RTS R5

2137
2138 025356 005037 003214 ADJCYL: CLR ADJTRK ;INSIDE/OUTSIDE TRACK FLAG
2139 025362 005037 003114      CLR HEAD01 ;INIT TO TOP SURFACE
2140 025366 012737 000001 003216      MOV #1,ADJUUT ;START OF TRACK LIST
2141 025374 012701 002236 21$: MOV #OUT10,R1 ;
2142 025400 012537 003100 20$: MOV (R5)+,ADJLOC ;PICK UP TRACK OFFSET
2143 025404 001003      BNE 1$ ;IS THERE ONE?
2144 025406 005037 003104      CLR ADJDIR
2145 025412 000205      RTS R5 ;NO EXIT
2146 025414 012537 003220 1$: MOV (R5)+,ADJLC2 ;YES, GET REST OF INFO
2147 025420 012537 003222      MOV (R5)+,ADJLC3
2148 025424 012537 003224      MOV (R5)+,ADJLC4
2149 025430 113700 003100 2$: MOVB ADJLOC,R0 ;GET OFFSET
2150 025434 012737 000020 003230      MOV #16.,STSEC ;STARTING SECTOR IS 16
2151
2152 025442 010102      MOV R1,R2 ;GET START INTO R2
2153
2154 025444 005300 3$: DEC R0 ;DOWN COUNT OFFSET
2155 025446 001414      BEQ 4$ ;FOUND IT?
2156
2157 025450 005722      TST (R2)+ ;INDEX (R2)
2158 025452 062737 000042 003230      ADD #34.,STSEC ;NO, NEXT SECTOR
2159 025460 022737 000050 003230      CMP #40.,STSEC
2160 025466 003366      BGT 3$
2161 025470 162737 000050 003230      SUB #40.,STSEC
2162 025476 000762      BR 3$ ;BACK FOR NEXT
2163
2164 025500 021227 177777 4$: CMP (R2),#-1 ;LEGAL TRACK?
2165 025504 001002      BNE 5$ ;YES, CONTINUE
2166
2167 025506 000137 026154      JMP 13$ ;NO PICK UP NEXT SET
2168
2169 025512 005037 003120 5$: CLR SURF ;SET UP FOR OUTER TRACK
2170 025516 005037 003122      CLR CYL
2171
2172 025522 005737 003114      TST HEAD01 ;WHICH HEAD?
2173 025526 001403      BEQ 6$ ;TOP, SKIP
2174
2175 025530 052737 000001 003120      BIS #1,SURF ;LOWER HEAD, SET IT!
2176
2177 025536 004537 026302 6$: JSR R5,SKCYL ;SEEK TO OUTER TRACK
2178

```

GLOBAL SUBROUTINES SECTION

```

2179 025542 011237 003122      MOV      (R2),CYL      ;GET DESIRED TRACK
2180
2181 025546 004537 026302      JSR      R5,SKCYL      ;SEEK TO IT
2182 025552 012737 020116 003116  MOV      #FWD,DIRC     ;SEEK DIRECTION
2183 025560 113703 003101      MOV      ADJLOC+1,R3   ;GET SECTORS TO WRITE
2184 025564 000303      SWAB     R3            ;ALIGN IT
2185 025566 042703 000377      BIC      #377,R3       ;CLEAR OUT HIGH BYTE
2186
2187 025572 022737 000047 003230  CMP      #39.,STSEC    ;OVER FORTY?
2188 025600 002003      BGE      7$           ;NO, CONTINUE
2189
2190 025602 162737 000050 003230  SUB      #40.,STSEC    ;YES BACK IT UP
2191 025610 013737 003230 025622 7$:     MOV      STSEC,8$      ;STARTING SECTOR
2192
2193 025616 004537 025116      JSR      R5,WRSEC      ;WRITE SECTORS
2194 025622 000000      .WORD   0
2195 025624 013737 025622 025636 8$:     MOV      8$,108$
2196 025632 004537 030140      JSR      R5,VAJWR      ;VERIFY THIS WRITE
2197 025636 000000      .WORD   0
2198 025640 013737 025636 025652 108$:   MOV      108$,208$
2199 025646 004537 030404      JSR      R5,BSVWR
2200 025652 000000      .WORD   0
2201 025654 013737 003230 003226 208$:   MOV      STSEC,STSEC1 ;GET OTHER SECTORS TO WRITE
2202 025662 062737 000010 003226  ADD      #8.,STSEC1    ;8 SECTORS GONE BY
2203 025670 022737 000047 003226  CMP      #39.,STSEC1  ;GONE PAST 40?
2204 025676 002003      BGE      9$           ;NO, OKAY
2205
2206 025700 162737 000050 003226  SUB      #40.,STSEC1  ;YES BACK IT UP
2207
2208 025706 013703 003220      9$:     MOV      ADJLC2,R3     ;GET SECTORS TO WRITE
2209
2210 025712 013737 003226 025724  MOV      STSEC1,10$   ;STARTING SECTORS
2211
2212 025720 004537 025116      JSR      R5,WRSEC      ;WRITE SECTORS
2213 025724 000000      .WORD   0
2214 025726 013737 025724 025740 10$:   MOV      10$,110$
2215 025734 004537 030140      JSR      R5,VAJWR      ;VERIFY THIS WRITE
2216 025740 000000      .WORD   0
2217 025742 013737 025740 025754 110$:   MOV      110$,210$
2218 025750 004537 030404      JSR      R5,BSVWR      ;VERIFY ADJ CYL + 1
2219 025754 000000      .WORD   0
2220 025756 022737 000001 003060 210$:   CMP      #1,T.DRIVE
2221 025764 001004      BNE      77$
2222 025766 012737 000377 003122  MOV      #377,CYL
2223 025774 000403      BR       88$
2224
2225 025776 012737 000777 003122 77$:   MOV      #777,CYL
2226
2227 026004 004537 026302      88$:   JSR      R5,SKCYL
2228
2229 026010 011237 003122      MOV      (R2),CYL     ;SEEK BACK TO PROPER TRACK
2230
2231 026014 004537 026302      JSR      R5,SKCYL     ;SEEK TO PROPER CYLINDER
2232 026020 012737 020126 003116  MOV      #REV,DIRC    ;SEEK DIRECTION
2233 026026 113703 003223  MOV      ADJLC3+1,R3  ;GET SECTORS TO WRITE
2234
2235 026032 000303      SWAB     R3            ;ALIGN IT

```

GLOBAL SUBROUTINES SECTION

```

2236 026034 042703 000377          BIC      #377,R3          ;CLEAR OUT HIGH BYTE
2237 026040 013737 003230 026052  MOV      STSEC,11$
2238
2239 026046 004537 025116          JSR      R5,WRSEC       ;WRITE PROPER SECTOR
2240 026052 000000          11$:    .WORD      0
2241
2242 026054 013737 026052 026066  MOV      11$,111$
2243 026062 004537 030140          JSR      R5,VAJWR       ;VERIFY THIS WRITE
2244 026066 000000          111$:   .WORD      0
2245 026070 013737 026066 026102  MOV      111$,211$
2246 026076 004537 030404          JSR      R5,BSVWR
2247 026102 000000          211$:   .WORD      0
2248 026104 013703 003224          MOV      ADJLC4,R3
2249 026110 013737 003226 026122  MOV      STSEC1,12$    ;GET SECTORS
;GET SECTORS TO WRITE
2250
2251 026116 004537 025116          JSR      R5,WRSEC       ;WRITE PROPER SECTORS
2252 026122 000000          12$:    .WORD      0
2253
2254 026124 013737 026122 026136  MOV      12$,112$
2255 026132 004537 030140          JSR      R5,VAJWR       ;VERIFY THIS WRITE
2256 026136 000000          112$:   .WORD      0
2257
2258 026140 013737 026136 026152  MOV      112$,212$
2259 026146 004537 030404          JSR      R5,BSVWR       ;VERIFY ADJ CYLINDERS + 1
2260 026152 000000          212$:   .WORD      0
2261
2262 026154 005737 003114          13$:    TST      HEAD01   ;WHICH HEAD WERE WE DOING?
2263 026160 001003          BNE      14$
2264 026162 005237 003114          INC      HEAD01
2265 026166 000402          BR       99$
2266 026170 005037 003114          14$:    CLR      HEAD01   ;NEXT SET OF TRACKS
2267 026174 062701 000012          99$:    ADD      #10.,R1     ;NEXT SET OF TRACKS
2268 026200 020127 002400          CMP      R1,#INN51    ;END OF LIST
2269 026204 002002          BGE      18$          ;END OF TRACK LIST
2270 026206 000137 025430          JMP      2$          ;NO GO BACK
2271
2272          ;AT END OF TRACK LIST NEXT GROUP OF WRITES
2273
2274 026212 005737 003064          18$:    TST      FADJ
2275 026216 001403          BEQ      15$          ;FIRST SET?
2276 026220 005037 003064          CLR      FADJ        ;NO, CONTINUE
2277 026224 000421          BR       17$        ;YES, CLEAR FIRST
2278 026226 005737 003214          15$:    TST      ADJTRK   ;EXIT
2279 026232 001004          BNE      16$        ;DONE BOTH INSIDE OUTSIDE
2280 026234 005237 003214          INC      ADJTRK     ;TRACKS, YES 16$
2281 026240 000137 025374          JMP      21$        ;NO, SET INSIDE FLAG
2282 026244 005037 003214          16$:    CLR      ADJTRK   ;GO DO INSIDE TRACK
2283 026250 005237 003216          INC      ADJUUT     ;BACK TO OUTSIDE TRACK
2284 026254 023737 003216 003130  CMP      ADJUUT,UUT  ;DONE WITH ANOTHER
2285 026262 001402          BEQ      17$        ;DONE TABLE FOR ALL UUT?
2286 026264 000137 025374          JMP      21$        ;YES, FOR EXIT
2287 026270 005725          17$:    TST      (R5)+    ;NO, GO BACK FOR NEXT
2288 026272 001376          BNE      17$        ;BUMP EXIT TO END OF
2289 026274 005037 003104          CLR      ADJDIR     ;TABLE FOR PROPER RETURN
2290 026300 000205          RTS      R5        ;EXIT
2291
2292          ;ROUTINE TO SEEK TO A DESIRED CYLINDER

```

GLOBAL SUBROUTINES SECTION

```

2293 ;CALL: JSR R5,SKCYL
2294 ;ROUTINE HAS DESIRED CYLINDER IN LOC "CYL"
2295 ;
2296 ;
2297 026302 010146 SKCYL: MOV R1,-(SP) ;SAVE R1
2298 026304 004537 032432 90$: JSR R5,LDFUNC ;GET PRESENT POSITION
2299 026310 000010 RDHDR
2300 ;
2301 026312 005737 003074 TST ERFLG ;ERROR FLAG SET
2302 026316 001104 BNE 5$ ;YES, SKIP
2303 ;
2304 026320 005001 CLR R1
2305 026322 012737 000007 003056 MOV #7,TEM
2306 026330 053701 003122 BIS CYL,R1 ;GET THE SELECTED CYLINDER NUMBER
2307 ;
2308 026334 006301 120$: ASL R1
2309 026336 005337 003056 DEC TEM
2310 026342 001374 BNE 120$
2311 026344 042737 000177 003172 BIC #177,E.MP
2312 026352 163701 003172 SUB E.MP,R1 ;CALCULATE DIFFERENCE WORD
2313 026356 103002 BCC 1$ ;IF POSITIVE SET DIRECTION
2314 026360 005401 NEG R1 ;NEGATE
2315 026362 000402 BR 2$ ;SKIP SETTING DIRECTION
2316 026364 052701 000004 1$: BIS #SIGN,R1 ;SET FOR FORWARD SEEK
2317 026370 052701 000001 2$: BIS #MK,R1 ;SET MARKER BIT
2318 026374 005737 003120 TST SURF
2319 026400 001402 BEQ 3$ ;TOP
2320 026402 052701 000020 BIS #SKHS,R1 ;BOTTOM
2321 026406 010137 003204 3$: MOV R1,BDA ;LOAD DIFFERENCE WORD
2322 026412 004537 032432 JSR R5,LDFUNC ;EXECUTE SEEK
2323 026416 000006 SEEK
2324 ;
2325 026420 005737 003074 TST ERFLG ;ERROR?
2326 026424 001041 BNE 5$ ;YES, SKIP
2327 ;
2328 026426 004537 032432 JSR R5,LDFUNC ;VERIFY POSITION?
2329 026432 000010 RDHDR
2330 026434 005737 003074 TST ERFLG
2331 026440 001033 BNE 5$
2332 026442 042737 000077 003172 BIC #77,E.MP ;VERIFY POSITION
2333 026450 005001 CLR R1
2334 026452 012737 000007 003056 MOV #7,TEM
2335 026460 053701 003122 BIS CYL,R1
2336 026464 006301 220$: ASL R1
2337 026466 005337 003056 DEC TEM
2338 026472 001374 BNE 220$
2339 026474 005737 003120 TST SURF
2340 026500 001402 BEQ 4$
2341 026502 052701 000100 BIS #HEAD,R1
2342 026506 020137 003172 4$: CMP R1,E.MP
2343 026512 001414 BEQ 6$
2344 ;
2345 026514 ERRDF 12.,SKER,ERR6 ;MIS SEEK ERROR
026514 104455 TRAP C$ERDF
026516 000014 .WORD 12
026520 020077 .WORD SKER
026522 020704 .WORD ERR6
    
```

GLOBAL SUBROUTINES SECTION

```

2346 026524 000137 026304          JMP      90$
2347
2348 026530          5$:  ERRDF   13.,FUNERR,ERR5 ;ERROR IN SEEK OPERATION
      026530 104455      TRAP    C$ERDF
      026532 000015      .WORD   13
      026534 020047      .WORD   FUNERR
      026536 020644      .WORD   ERR5
2349 026540 000137 026304          JMP      90$
2350 026544 012601      6$:  MOV     (SP)+,R1      ;CANT GET THERE
2351 026546 000205      RTS     R5              ;EXIT
2352
2353          ;ROUTINE TO PERFORM REGISTER PRINTOUT DUMP
2354          ;CALL: JSR     PC,REGDMP
2355          ;PROMPT - BEFORE CS: _ BA: _ DA: _ MP: _
2356 026550      REGDMP: PRINTB  #FRM12,BCS,BBA,BDA,BMP
      026550 013746 003206      MOV     BMP,-(SP)
      026554 013746 003204      MOV     BDA,-(SP)
      026560 013746 003202      MOV     BBA,-(SP)
      026564 013746 003200      MOV     BCS,-(SP)
      026570 012746 021646      MOV     #FRM12,-(SP)
      026574 012746 000005      MOV     #5,-(SP)
      026600 010600      MOV     SP,R0
      026602 104414      TRAP    C$PNTB
      026604 062706 000014      ADD     #14,SP
2357
2358 026610          ;PROMPT - AFTER CS: _ BA: _ DA: _ MP: _
      026610 013746 003172      PRINTB #FRM13,E.CS,E.BA,E.DA,E.MP
      026614 013746 003170      MOV     E.MP,-(SP)
      026620 013746 003166      MOV     E.DA,-(SP)
      026624 013746 003164      MOV     E.BA,-(SP)
      026630 012746 021725      MOV     E.CS,-(SP)
      026634 012746 000005      MOV     #FRM13,-(SP)
      026640 010600      MOV     #5,-(SP)
      026642 104414      MOV     SP,R0
      026644 062706 000014      TRAP    C$PNTB
      026644 062706 000014      ADD     #14,SP
2359 026650 032737 040000 003164      BIT     #BIT14,E.CS
2360 026656 001437      BEQ     1$
2361 026660 016403 000000      MOV     CSR(R4),R3
2362 026664 012763 000013 000004      MOV     #13,DA(R3)
2363 026672 012737 000004 003200      MOV     #4,BCS
2364 026700 056437 000004 003200      BIS     DSB(R4),BCS
2365 026706 013763 003200 000000      MOV     BCS,CS(R3)
2366 026714 032763 000200 000000 2$:  BIT     #200,CS(R3)
2367 026722 001774      BEQ     2$
2368 026724 016337 000006 003106      MOV     MP(R3),DRSTAT
2369 026732          ;PROMPT - DRIVE STATUS
      026732 013746 003106      PRINTB #FRM14,DRSTAT
      026736 012746 022010      MOV     DRSTAT,-(SP)
      026742 012746 000002      MOV     #FRM14,-(SP)
      026746 010600      MOV     #2,-(SP)
      026750 104414      MOV     SP,R0
      026752 062706 000006      TRAP    C$PNTB
2370 026756 000207      ADD     #6,SP
2371          1$:  RTS     PC
2372
2373          ;ROUTINE TO STORE OR RETRIEVE ADJACENT CYLINDER SECTOR DRIVE
2374          ;INFORMATION FROM THE 24X5 "SECLST" BUFFER.
          ;ENTER WITH R0 = SECTOR REQUEST

```

GLOBAL SUBROUTINES SECTION

```

2375      ;EXIT WITH R0 = ADJACENT CYLINDER DRIVE INFORMATION FOR SECTOR
2376      ;EXIT WITH R0 = 0 IF SECTOR REQUESTED IS NOT IN BUFFER MAP
2377      ;CALL 1:      JSR R5,RSADJS
2378      ;              .WORD 0              ;RETRIEVE SECTOR INFO.
2379      ;CALL 2:      JSR R5,RSADJS
2380      ;              .WORD 1              ;STORE SECTOR INFO.
2381 026760 010146      RSADJS: MOV      R1,-(SP)
2382 026762 010246      MOV      R2,-(SP)
2383 026764 010346      MOV      R3,-(SP)
2384 026766 042700 177700      BIC      @177700,R0      ;SAVE SECTOR BITS
2385 026772 012537 003102      MOV      (R5)+,ADJFLG   ;SAVE RETRIEVE/STORE FLAG
2386 026776 012701 000001      MOV      @1,R1          ;START WITH TRACK (N-2)
2387 027002 012702 002442      MOV      @SECBUF,R2     ;START OF 24X5 BUFFER
2388 027006 012703 000020      MOV      @16.,R3        ;SECTOR 16 START FOR (N-2) TRACK
2389 027012 123701 003100      1@:    CMPB     ADJLOC,R1  ;CHECK TRACK INDEX
2390 027016 001413      BEQ      2@
2391 027020 005201      INC      R1              ;INDEX TRACK REFERENCE
2392 027022 062702 000060      ADD      @48.,R2         ;UPDATE BUFFER TO NEXT TRACK REF.
2393 027026 062703 000042      ADD      @34.,R3         ;UPDATE SECTOR START FOR NEXT TRACK
2394 027032 020327 000050      CMP      R3,@40.
2395 027036 002765      BLT      1@
2396 027040 162703 000050      SUB      @40.,R3
2397 027044 000762      BR       1@
2398 027046 012701 000030      2@:    MOV      @24.,R1      ;SET COUNTER FOR 24 SECTORS
2399 027052 020003      3@:    CMP      R0,R3        ;COMPARE SECTOR TO SECTOR TABLE
2400 027054 001413      BEQ      5@              ;YES,STORE OR RETRIEVE SECTOR INFO.
2401 027056 005722      TST      (R2)+           ;INDEX SECLST BUFFER IN WORD FORMAT
2402 027060 005203      INC      R3              ;INDEX SECTOR COUNT
2403 027062 020327 000047      CMP      R3,@39.        ;COMPARE SECTOR COUNT FOR <40
2404 027066 003402      BLE      4@
2405 027070 162703 000050      SUB      @40.,R3        ;KEEP SECTOR COUNT<40
2406 027074 005301      4@:    DEC      R1          ;PASSED 24 SECTORS?
2407 027076 001365      BNE      3@              ;COMPARE NEXT SECTOR
2408 027100 005000      CLR      R0              ;SETUP R0 FOR EXIT
2409 027102 000405      BR       7@              ;EXIT ROUTINE,SECTOR NOT FOUND
2410 027104 005737 003102      5@:    TST      ADJFLG     ;FLAG=0 FOR RETRIEVE
2411 027110 001401      BEQ      6@
2412 027112 010412      MOV      R4,(R2)         ;STORE DRIVE INFO. INTO BUFFER
2413 027114 011200      6@:    MOV      (R2),R0     ;SAVE DRIVE INFO. INTO R0 FOR EXIT
2414 027116 012603      7@:    MOV      (SP)+,R3
2415 027120 012602      MOV      (SP)+,R2
2416 027122 012601      MOV      (SP)+,R1
2417 027124 000205      RTS      R5              ;EXIT
2418
2419      ;ROUTINE TO SET DRIVE IN SECTOR LIST
2420      ;CALL: JSR R5,SETLST ;R0 HAS SECTOR
2421      ;DRIVE GOTTEN FROM R4
2422
2423 027126 010146      SETLST: MOV      R1,-(SP) ;SAVE R1
2424
2425 027130 162700 000034      SUB      @28.,R0        ;START LIST AT 0
2426 027134 100002      BPL      3@
2427 027136 062700 000050      ADD      @40.,R0
2428 027142 012701 002402      3@:    MOV      @SECLST,R1 ;BEGINNING OF SECTOR LIST
2429 027146 005700      1@:    TST      R0          ;FOUND SECTOR?
2430 027150 001403      BEQ      2@              ;BRANCH IF YES
2431 027152 005300      DEC      R0              ;DECREMENT SECTOR

```

GLOBAL SUBROUTINES SECTION

```

2432 027154 005721          TST      (R1)+      ;NEXT ENTRY IN LIST
2433 027156 000773          BR       1$         ;GO BACK
2434 027160 010411          2$:     MOV      R4,(R1)      ;STORE DRIVE BITS IN LIST
2435 027162 012601          MOV      (SP)+,R1    ;RESTORE R1
2436 027164 000205          RTS      R5
2437
2438          ;ROUTINE TO LOCATE DRIVE THAT WROTE SECTOR LAST
2439          ;CALL: JSR      R5,FNDDRV      ;R0-CONTAINS SECTOR
2440          ;ON EXIT R0-DRIVE
2441
2442 027166 010146          FNDDRV: MOV     R1,-(SP)      ;SAVE R1
2443 027170 162700 000034    SUB     #28.,R0      ;START LIST AT 0
2444 027174 100002          BPL     3$         ;
2445 027176 062700 000050    ADD     #40.,R0      ;
2446 027202 012701 002402    3$:     MOV     #SECLST,R1    ;START OF LIST
2447 027206 005700          1$:     TST     R0         ;FOUND SECTOR?
2448 027210 001403          BEQ     2$         ;YES, GET DRIVE #, EXIT
2449 027212 005300          DEC     R0         ;NO, DOWN COUNT SECTOR
2450 027214 005721          TST     (R1)+      ;NEXT ENTRY IN LIST
2451 027216 000773          BR       1$         ;GO BACK
2452 027220 011100          2$:     MOV     (R1),R0      ;GET DRIVE BUFFER POINTER
2453 027222 012601          MOV     (SP)+,R1    ;RESTORE R1
2454 027224 000205          RTS      R5         ;EXIT
2455
2456          ;
2457          ;ROUTINE TO VERIFY THAT THE OVERWRITE DID ACTUALLY OVERWRITE THE
2458          ;PREVIOUS DATA ON THE PACK.
2459          ;
2460          ;CALL: JSR      R5,VEROW      USES R3 AS BIT MAP OF SECTORS TO
2461          ;                               CHECK. R3 IS LOADED PRIOR TO
2462          ;                               WRITING SECTORS.
2463          ;
2464 027226 010046          VEROW:  MOV     R0,-(SP)      ;SAVE REGISTER CONTENTS
2465 027230 010146          MOV     R1,-(SP)
2466 027232 010246          MOV     R2,-(SP)
2467 027234 012737 000034 003132    MOV     #28.,SECT    ;START VERIFY AT SECTOR 28
2468 027242 012701 100000          MOV     #100000,R1   ;BIT MASK FOR VERIFICATION
2469 027246 016437 000006 003136    MOV     PAT(R4),GDATA ;GET PATTERN FOR THIS DRIVE
2470
2471 027254 012737 177600 003206 1$:     MOV     #-128.,BMP    ;SET UP READ-ONE SECTOR
2472 027262 012737 003232 003202    MOV     #BUF,BBA     ;BUS ADDRESS
2473 027270 042737 000077 003204    2$:     BIC     #77,BDA     ;CLEAR OUT SECTOR BITS
2474 027276 053737 003132 003204    BIS     SECT,BDA     ;SET SECTOR
2475 027304 030103          BIT     R1,R3       ;DO WE READ THIS ONE?
2476 027306 001521          BEQ     5$         ;NO, BRANCH
2477 027310 004537 032432          JSR     R5,LDFUNC    ;READ
2478 027314 000014          READ
2479
2480 027316 005737 003164          TST     E.CS        ;ERROR
2481 027322 100107          BPL     4$         ;NO CONTINUE
2482
2483 027324 005737 003062          TST     F0WR        ;INITIAL WRITE
2484 027330 001412          BEQ     21$        ;NO
2485 027332 012737 017373 003072    MOV     #INITWR,REASON ;SETUP INITIAL WRITE OF SECTOR
2486 027340 016437 000000 003070    MOV     CSR(R4),LSTCLR
2487 027346 016437 000005 003134    MOV     DSB+1(R4),LSTDRV
2488 027354 000415          BR      22$

```


GLOBAL SUBROUTINES SECTION

```

2489 027356 012737 017642 003072 21:  MOV    #OVWES,REASON    ;SET MESSAGE FOR OVERWRITE
2490 027364 013700 003132          MOV    SECT,R0          ;FIND DRIVE THAT LAST WROTE
2491 027370 004537 027166          JSR    R5,FNDDRV       ;SECTOR
2492 027374 016037 000000 003070  MOV    CSR(R0),LSTCLR   ;GET IT'S CSR
2493 027402 116037 000005 003134  MOVB   DSB+1(R0),LSTDRV ;GET THE DRIVE
2494 027410          22:  ERDF   13.,OVWER,ERR4   ;PRINT ERROR
      027410 104455          TRAP   C#ERDF
      027412 000015          .WORD 13
      027414 020003          .WORD OVWER
      027416 020464          .WORD ERR4
2495 027420 005037 003142          CLR    WCOUNT        ;CLEAR BAD WORD COUNT W/IN SECTOR
2496 027424 005037 003144          CLR    SECWRD         ;CLEAR WORD IN SECTOR
2497 027430 012702 003232          MOV    #BUF,R2        ;GET BUFFER START
2498 027434 023712 003136          3:  CMP    GDATA,(R2)     ;IS DATA CORRECT?
2499 027440 001417          BEQ    31:            ;YES CHECK NEXT
2500 027442 005237 003142          INC    WCOUNT        ;NO ACCOUNT FOR IT
2501 027446          PRINTF #FRM8,SECWRD,GDATA,(R2)
      027446 011246          MOV    (R2),-(SP)
      027450 013746 003136          MOV    GDATA,-(SP)
      027454 013746 003144          MOV    SECWRD,-(SP)
      027460 012746 021437          MOV    #FRM8,-(SP)
      027464 012746 000004          MOV    #4,-(SP)
      027470 010600          MOV    SP,R0
      027472 104417          TRAP   C#PNTF
      027474 062706 000012          ADD    #12,SP
2502
2503 027500 005722          31:  TST    (R2),          ;NEXT
2504 027502 005237 003144          INC    SECWRD         ;NEXT
2505 027506 023727 003144 000200  CMP    SECWRD,#128.   ;DONE WITH SECTOR?
2506 027514 001347          BNE    3:            ;NO GO BACK
2507
2508 027516          PRINTF #FRM9,WCOUNT  ;PRINT SUMMARY
      027516 013746 003142          MOV    WCOUNT,-(SP)
      027522 012746 021503          MOV    #FRM9,-(SP)
      027526 012746 000002          MOV    #2,-(SP)
      027532 010600          MOV    SP,R0
      027534 104417          TRAP   C#PNTF
      027536 062706 000006          ADD    #6,SP
2509
2510 027542 013700 003132          4:  MOV    SECT,R0        ;SET SECTOR IN LIST TO THE
2511 027546 004537 027126          JSR    R5,SETLST      ;CREDIT OF THIS DRIVE
2512
2513 027552 005237 003132          5:  INC    SECT          ;NEXT SECTOR
2514 027556 023727 003132 000050  CMP    SECT,#40.
2515 027564 001003          BNE    6:
2516 027566 162737 000050 003132  SUB    #40.,SECT
2517 027574 000241          6:  CLC                    ;CLEAR CARRY
2518 027576 006001          ROR                    ;NEXT BIT
2519 027600 103225          BCC    1:            ;IF CLEAR NEXT
2520
2521 027602 012602          MOV    (SP),R2        ;RESTORE R2-RO, EXIT
2522 027604 012601          MOV    (SP),R1
2523 027606 012600          MOV    (SP),R0
2524 027610 000205          RTS    R5
2525
2526          ;ROUTINE TO VERIFY THAT A DRIVE CAN RECOVER ANOTHER DRIVE'S DATA.
2527          ;

```

GLOBAL SUBROUTINES SECTION

```

2528 ;CALL: JSR R5,VEROD USES R3 AS BIT MAP OF SECTORS TO
2529 ; CHECK. R3 IS LOAD BY WRSEC (WE
2530 ; USE R3 COMPLIMENTED.
2531 ;
2532 ;
2533 VEROD: MOV R0,-(SP) ;SAVE R0-R2
2534 MOV R1,-(SP)
2535 MOV R2,-(SP)
2536 MOV #100000,R1 ;BIT MASK FOR SECTORS
2537 MOV #28.,SECT ;START WITH SECTOR 28
2538 TST F0WR ;CHECK FOR FIRST OVERWRITE
2539 BNE 6$
2540
2541 1$: MOV #128.,BMP ;SET UP READ (ONE SECTOR)
2542 MOV #BUF,BBA ;BUS ADDRESS
2543 2$: BIC #77,BDA ;CLEAR SECTOR BITS
2544 BIS SECT,BDA ;SET IN SECTOR BITS
2545 BIT R1,R3 ;CHECK THIS SECTOR?
2546 BNE 5$ ;NO BRANCH
2547
2548 MOV SECT,R0 ;FIND DRIVE THAT WROTE
2549 JSR R5,FNDDRV ;SECTOR LAST
2550 MOV CSR(R0),LSTCLR ;GET CSR OF DRIVE
2551 MOVB DSB+1(R0),LSTDRV ;GET DRIVE
2552 MOV PAT(R0),GDATA ;GET PATTERN
2553
2554 JSR R5,LDFUNC ;READ
2555 READ
2556
2557 TST E.CS ;ERROR?
2558 BPL 5$ ;NO, NEXT SECTOR
2559 MOV #RECHS,REASON ;SET READ RECOVERY MESSAGE
2560 ERRDF 14.,RECER,ERR4 ;REPORT ERROR
2561 TRAP C#ERDF
2562 .WORD 14
2563 .WORD RECER
2564 .WORD ERR4
2565
2566 CLR WCOUNT ;CLEAR BAD WORD COUNT
2567 CLR SECWRD ;CLEAR WORD W/I SECTOR
2568 MOV #BUF,R2 ;START OF BUFFER
2569 3$: CMP GDATA,(R2) ;DATA COMPARE
2570 BEQ 4$ ;YES, CHECK NEXT
2571
2572 INC WCOUNT ;ACCOUNT FOR ERROR
2573 PRINTF #FRMB,SECWRD,GDATA,(R2) ;PRINT ERROR
2574 MOV (R2),-(SP)
2575 MOV GDATA,-(SP)
2576 MOV SECWRD,-(SP)
2577 MOV #FRMB,-(SP)
2578 MOV #4,-(SP)
2579 MOV SP,R0
2580 TRAP C#PNTF
2581 ADD #12,SP
2582
2583 4$: TST (R2)+ ;NEXT
2584 INC SECWRD ;NEXT WORD IN SECTOR

```

GLOBAL SUBROUTINES SECTION

```

2573 030046 023727 003144 000200      CMP      SECWRD,#128.      ;DONE?
2574 030054 001347                    BNE      3$                ;NO
2575 030056                    PRINTF   #FRM9,WCOUNT      ;PRINT SUMMARY
                030056 013746 003142      MOV      WCOUNT,-(SP)
                030062 012746 021503      MOV      #FRM9,-(SP)
                030066 012746 000002      MOV      #2,-(SP)
                030072 010600      MOV      SP,R0
                030074 104417      TRAP    C$PNTF
                030076 062706 000006      ADD      #6,SP

2576                    5$:      INC      SECT                ;NEXT SECTOR
2577 030102 005237 003132 000050      CMP      SECT,#40.
2578 030106 023727 003132                    BNE      7$
2579 030114 001002                    CLR      SECT
2580 030116 005037 003132
2581 030122 000241                    7$:      CLC
2582 030124 006001                    ROR      R1                ;NEXT BIT MAP
2583 030126 103244                    BCC      1$
2584
2585 030130 012602                    6$:      MOV      (SP)+,R2        ;RESTORE R2-R0, EXIT
2586 030132 012601                    MOV      (SP)+,R1
2587 030134 012600                    MOV      (SP)+,R0
2588 030136 000205                    RTS      R5
2589
2590                    ;ROUTINE TO VERIFY THE ADJ. CYL. WRITE IS GOOD
2591                    ;USES R3 AND WORD FOLLOWING CALL
2592                    ;IF WRITE WAS GOOD,SECTOR WILL BE STORED IN MAP
2593                    ;USING RSADJS/.WORD 1
2594
2595 030140 010046                    VAJWR:  MOV      R0,-(SP)        ;SAVE REGISTERS
2596 030142 010146                    MOV      R1,-(SP)
2597 030144 010246                    MOV      R2,-(SP)
2598 030146 012701 100000      MOV      #100000,R1      ;BIT MASK FOR CYLINDER
2599 030152 012502                    MOV      (R5)+,R2      ;STARTING SECTOR
2600 030154 005000                    CLR      R0
2601 030156 053700 003122      BIS      CYL,R0
2602 030162 012737 000007 003056      MOV      #7,TEM
2603
2604 030170 006300                    2$:      ASL      R0
2605 030172 005337 003056      DEC      TEM
2606 030176 001374                    BNE      2$
2607 030200 005737 003120      TST      SURF
2608 030204 001402                    BEQ      3$
2609 030206 052700 000100      BIS      #HEAD,R0
2610 030212 050200                    3$:      BIS      R2,R0
2611 030214 030103                    4$:      BIT      R1,R3
2612 030216 001462                    BEQ      5$
2613 030220 012737 177600 003206      MOV      #-128.,BMP
2614 030226 010037 003204      MOV      R0,BDA
2615 030232 010037 003066      MOV      R0,TEMP
2616 030236 042700 177700      BIC      #177700,R0
2617 030242 020027 000047      CMP      R0,#39.
2618 030246 003406                    BLE      6$
2619 030250 162737 000050 003204      SUB      #40.,BDA
2620 030256 162737 000050 003066      SUB      #40.,TEMP
2621 030264 012737 003232 003202 6$:      MOV      #BUF,BBA
2622 030272 005037 003110      CLR      HSFLG
2623 030276 013700 003066      MOV      TEMP,R0
    
```

GLOBAL SUBROUTINES SECTION

```

2624 030302 004537 032432      10$: JSR    R5,LDFUNC      ;READ FUNCTION
2625 030306 000014              READ
2626 030310 005737 003074      TST    ERFLG
2627 030314 001416              BEQ    7$
2628 030316 005737 003110      TST    HSFLG
2629 030322 001007              BNE    11$
2630 030324              ERRSOFT 120.,READ1,ERR2
      030324 104457              TRAP  C$ERSOFT
      030326 000170              .WORD 120
      030330 020163              .WORD READ1
      030332 020276              .WORD ERR2
2631 030334 005237 003110      INC    HSFLG
2632 030340 000760              BR     10$
2633 030342              11$: ERRHRD 130.,READ1,ERR2
      030342 104456              TRAP  C$ERHRD
      030344 000202              .WORD 130
      030346 020163              .WORD READ1
      030350 020276              .WORD ERR2
2634 030352 010046              7$:  MOV    RO,-(SP)
2635 030354 004537 026760      JSR    R5,RSADJS      ;STORE ADJ. CYL. SECTOR INFO.
2636 030360 000001              .WORD 1
2637 030362 012600              MOV    (SP)+,RO      ;RESTORE RO
2638 030364 005200              5$:  INC    RO
2639 030366 000241              CLC
2640 030370 006001              ROR    R1
2641 030372 103310              BCC    4$
2642 030374 012602              MOV    (SP)+,R2      ;RESTORE REGISTERS AND EXIT
2643 030376 012601              MOV    (SP)+,R1
2644 030400 012600              MOV    (SP)+,RO
2645 030402 000205              RTS    R5
2646
2647 ;ROUTINE TO VERIFY THAT WRITE DID NOT DISTURB ADJACENT TRACKS
2648 ;WRITTEN BY OTHER DRIVES.
2649 ;CALL JSR R5,BSVWR
2650 ;
2651 ;
2652 ;
2653 ;USES "ADJLOC" TO GET +1/-1 CYLINDER OFFSET
2654 ;USES R3 FOR SECTOR MAP, USES MAP AT "SECBUF" FOR INFO
2655 030404 010046      BSVWR: MOV    RO,-(SP)      ;SAVE REGISTERS
2656 030406 010146      MOV    R1,-(SP)
2657 030410 010246      MOV    R2,-(SP)
2658 030412 013746 003122      MOV    CYL,-(SP)
2659 030416 013746 003120      MOV    SURF,-(SP)
2660 030422 012546      MOV    (R5)+,-(SP)      ;GET STARTING SECTOR
2661 030424 123727 003100 000003      CMPB  ADJLOC,#3      ;ON MIDDLE TRACK???
2662 030432 001455      BEQ    BSEXIT        ;YES, THEN NO CHECK
2663 030434 162716 000042      SUB    #34.,(SP)      ;SETUP SECTOR START FOR OUTSIDE
2664 030440 100002      BPL    1$            ;IF POSITIVE OKAY ELSE FIX
2665 030442 062716 000050      ADD    #40.,(SP)      ;FIX IT
2666 030446 123727 003100 000001 1$:  CMPB  ADJLOC,#1      ;ON OUTER LIMIT???
2667 030454 001412      BEQ    INAWR        ;YES,SKIP CHECK
2668 030456 105337 003100      DECB  ADJLOC        ;OUTER ADJ TRACK
2669 030462 005337 003122      DEC    CYL
2670 030466 004537 030614      JSR    R5,CHECK      ;GO CHECK ADJ SECTORS
2671 030472 005237 003122      INC    CYL          ;FIX BACK
2672 030476 105237 003100      INCB  ADJLOC

```

GLOBAL SUBROUTINES SECTION

```

2673 030502 062716 000104          INAWR: ADD    #68.,(SP)          ;INNER SECTOR START
2674 030506 021627 000050          CMP    (SP),#40.          ;WITHIN LIMITS???
2675 030512 002407                   BLT    1$                  ;YES, OKAY
2676 030514 162716 000050          SUB    #40.,(SP)          ;FIX SECTOR
2677 030520 021627 000050          CMP    (SP),#40.
2678 030524 002402                   BLT    1$
2679 030526 162716 000050          SUB    #40.,(SP)
2680 030532 123727 003100 000005 1$:  CMPB  ADJLOC,#5          ;INNER LIMIT??
2681 030540 001412                   BEQ    BSEXIT             ;YES,SKIP CHECK
2682 030542 105237 003100          INCB  ADJLOC             ;FIX FOR INNER
2683 030546 005237 003122          INC   CYL                ;
2684 030552 004537 030614          JSR   R5,CHECK           ;GO CHECK ADJ SECTORS
2685 030556 105337 003100          DECB  ADJLOC             ;FIX BACK
2686 030562 005337 003122          DEC   CYL                ;
2687 030566 005726                   BSEXIT: TST (SP)+         ;THROW OFF SECTOR
2688 030570 012637 003120          MOV   (SP)+,SURF
2689 030574 012637 003122          MOV   (SP)+,CYL
2690 030600 012602          NCHECK: MOV (SP)+,R2
2691 030602 012601          MOV   (SP)+,R1
2692 030604 012600          MOV   (SP)+,R0
2693 030606 004537 026302          JSR   R5,SKCYL          ;SEEK BACK
2694 030612 000205                   RTS    R5                ;RETURN
2695
2696          ;ROUTINE TO VERIFY AN ADJACENT SECTOR
2697          ;CALLED FROM BSVWR
2698          ;CALL JSR R5,CHECK
2699          ;
2700
2701 030614 012701 100000          CHECK: MOV #100000,R1     ;SECTOR MASK
2702 030620 004537 026302          JSR   R5,SKCYL          ;GET TO DESIRED CYLINDER
2703 030624 005002                   CLR   R2                 ;CREATE ADDRESS
2704 030626 053702 003122          BIS   CYL,R2
2705 030632 012737 000007 003056 2$:  MOV   #7,TEM
2706 030640 006302                   ASL   R2
2707 030642 005337 003056          DEC   TEM
2708 030646 001374                   BNE   2$
2709 030650 005737 003120          TST   SURF
2710 030654 001402                   BEQ   3$                 ;NO
2711 030656 052702 000100          BIS   #HEAD,R2
2712 030662 056602 000002          3$:  BIS   2(SP),R2          ;SET IN SECTOR
2713 030666 030103          4$:  BIT   R1,R3            ;THIS SECTOR IN LIST???
2714 030670 001452                   BEQ   5$                 ;NO, NEXT
2715 030672 010200                   MOV   R2,R0              ;COPY SECTOR
2716 030674 042700 177700          BIC   #177700,R0        ;ONLY SECTOR LEFT
2717 030700 020027 000050          CMP   R0,#40.           ;SECTOR OKAY???
2718 030704 002404                   BLT   6$                 ;YES
2719 030706 162700 000050          SUB   #40.,R0
2720 030712 162702 000050          SUB   #40.,R2           ;FIX SECTOR
2721 030716 004537 026760          6$:  JSR   R5,RSADJS        ;FIND IF SECTOR PREVIOUSLY WRITTEN
2722 030722 000000                   .WORD 0
2723 030724 005700                   TST   R0                 ;WAS IT??
2724 030726 001433                   BEQ   5$                 ;NO
2725 030730 010237 003204          MOV   R2,BDA            ;LOAD DISK ADDRESS
2726 030734 012737 177600 003206  MOV   #-128.,BMP        ;LOAD WC
2727 030742 004537 032432          JSR   R5,LDFUNC         ;LOAD
2728 030746 000014                   READ
2729 030750 005737 003074          TST   ERFLG             ;WAS READ GOOD

```

GLOBAL SUBROUTINES SECTION

```

2730 030754 001420          BEQ      5$
2731 030756 010346          MOV      R3,-(SP)
2732 030760 010237 003132  MOV      R2,SECT
2733 030764 010003          MOV      R0,R3
2734 030766 042737 177700 003132 BIC      @177700,SECT
2735 030774          ERRHRD  140.,ADJTXT,ERR3
      030774 104456          TRAP    C$ERHRD
      030776 000214          .WORD  140
      031000 020210          .WORD  ADJTXT
      031002 020336          .WORD  ERR3
2736 031004 012603          MOV      (SP)+,R3
2737 031006          ERRHRD  110.,READ1,ERR2
      031006 104456          TRAP    C$ERHRD
      031010 000156          .WORD  110
      031012 020163          .WORD  READ1
      031014 020276          .WORD  ERR2
2738 031016 005202          5$: INC      R2          ;NEXT SECTOR
2739 031020 000241          CLC
2740 031022 006001          ROR      R1          ;SHIFT MASK
2741 031024 103320          BCC      4$
2742 031026 000205          RTS      R5
2743
2744          ;ROUTINE TO MERGE BAD SECTOR FILES
2745          ;ENTRY INTO THIS ROUTINE WILL OCCUR AFTER THE "SERNUM" ROUTINE
2746          ;IS PERFORMED. THE FACTORY BAD SECTOR FILE WILL BE LOCATED IN
2747          ;FIRST 400(8) LOCATIONS.
2748          ;THIS ROUTINE WILL STORE THE FIELD BAD SECTORS INTO THE NEXT
2749          ;400 LOCATIONS AND THEN MERGE THE FACTORY BAD FILE
2750          ;WITH THE FIELD BAD FILE.
2751
2752          ;FACTORY BAD AT BUF
2753          ;FIELD BAD AT BUF + 512.
2754
2755 031030 010146          MERGE:  MOV      R1,-(SP)          ;SAVE R1, R2, R3
2756 031032 010246          MOV      R2,-(SP)
2757 031034 010346          MOV      R3,-(SP)
2758 031036 012737 003632 003202  MOV      @BUF+400,BBA          ;BUFFER START FOR FIELD BAD
2759 031044 022737 000001 003060  CMP      @1,T.DRIVE
2760 031052 001004          BNE      55$
2761 031054 012737 077724 003204  MOV      @77724,BDA
2762 031062 000403          BR       66$
2763 031064 012737 177724 003204  55$: MOV      @177724,BDA
2764
2765 031072 012737 177400 003206  66$: MOV      @-256.,BMP
2766 031100 004537 032432          97$: JSR      R5,LDFUNC          ;LOAD READ FUNCTION
2767 031104 000014          READ
2768 031106 005737 003074          TST      ERFLG          ;TEST ERROR FLAG
2769 031112 001431          BEQ      98$          ;YES;MERGE BAD SECTOR FILES
2770 031114 062737 000004 003204  ADD      @4,BDA          ;TRY NEXT FIELD BAD SECTOR FILE
2771 031122 022737 000001 003060  CMP      @1,T.DRIVE
2772 031130 001004          BNE      400$
2773 031132 022737 077750 003204  CMP      @77750,BDA
2774 031140 001357          BNE      97$
2775
2776 031142 022737 177750 003204  400$: CMP      @177750,BDA
2777 031150 001353          BNE      97$          ;NO,DO NEXT FIELD BAD SECTOR
2778 031152          PRINTF @FRM15

```

GLOBAL SUBROUTINES SECTION

031152	012746	022037			MOV	#FRM15,-(SP)	
031156	012746	000001			MOV	#1,-(SP)	
031162	010600				MOV	SP,R0	
031164	104417				TRAP	C\$PNTF	
031166	062706	000004			ADD	#4,SP	
2779	031172			999\$:	BREAK		
	031172	104422			TRAP	C\$BRK	
2780	031174	000776			BR	999\$	
2781	031176	012701	003242	98\$:	MOV	#BUF+10,R1	;GET PAST ID ETC.
2782	031202	012702	000176		MOV	#126.,R2	;MAX = 126
2783	031206	005721		1\$:	TST	(R1)+	;SECTOR OR END
2784	031210	100404			BMI	2\$;END, GO GET FIELD
2785	031212	005721			TST	(R1)+	;REST OF SECTOR
2786	031214	005302			DEC	R2	;MAX REACHED
2787	031216	001373			BNE	1\$;NO, KEEP GOING
2788	031220	000401			BR	3\$;YES, SKIP BACK UP
2789	031222	005741		2\$:	TST	-(R1)	;BACK UP PAST TERMINATOR
2790	031224	012703	000176	3\$:	MOV	#126.,R3	;SET 126 MAX
2791	031230	012702	003642		MOV	#BUF+410,R2	;GET FIELD SECTORS
2792	031234	012221		4\$:	MOV	(R2)+,(R1)+	;MERGE AT END OF FACTORY
2793	031236	100403			BMI	5\$;DONE?
2794	031240	012221			MOV	(R2)+,(R1)+	;NO, MERGE REST OF SECTOR
2795	031242	005303			DEC	R3	;DONE
2796	031244	001373			BNE	4\$;NO, GO BACK
2797	031246	012603		5\$:	MOV	(SP)+,R3	;RESTORE R3, R2, R1
2798	031250	012602			MOV	(SP)+,R2	
2799	031252	012601			MOV	(SP)+,R1	
2800	031254	000205			RTS	R5	;EXIT
2801							
2802	031256	012537	003146	FNDTRK:	MOV	(R5)+,OFFSET	;GET INCREMENT/DECREMENT
2803	031262	012537	003156		MOV	(R5)+,SURFACE	;GET HEAD (SURFACE)
2804	031266	022737	000001 003060		CMP	#1,T.DRIVE	
2805	031274	001001			BNE	80\$	
2806	031276	000401			BR	90\$	
2807	031300	022525		80\$:	CMP	(R5)+,(R5)+	
2808	031302	012537	003152	90\$:	MOV	(R5)+,FRTTRK	
2809	031306	012537	003150		MOV	(R5)+,LSTTRK	
2810	031312	005037	003160		CLR	TRKFND	;CLEAR OUT FLAG FOUND
2811	031316	005037	003162		CLR	TRKCNT	;CLEAR OUT TRACK COUNT
2812	031322	013737	003152 003154		MOV	FRTTRK,PRSTRK	;GET FIRST TRACK
2813	031330			1\$:			
2814	031330	004537	031430		JSR	R5,FNDBSC	;IS TRACK IN BAD SECTOR FILE
2815	031334	005737	002234		TST	HDRFND	;WAS IT?
2816	031340	001003			BNE	2\$;YES, CLEAR TRKCNT
2817	031342	005237	003162		INC	TRKCNT	;NO, INDICATE GOOD TRACK
2818	031346	000402			BR	3\$;CONTINUE
2819	031350	005037	003162	2\$:	CLR	TRKCNT	;START COUNT OVER
2820	031354	023727	003162 000005	3\$:	CMP	TRKCNT,#5	;FIND 5 TRACKS YET?
2821	031362	001011			BNE	4\$;NO, CONTINUE
2822	031364	005237	003160		INC	TRKFND	;YES, EXIT WITH GOOD FLAG
2823	031370	022737	000001 003060		CMP	#1,T.DRIVE	
2824	031376	001002			BNE	81\$	
2825	031400	062705	000004		ADD	#4,R5	
2826							
2827	031404	000205		81\$:	RTS	R5	
2828	031406	023737	003154 003150	4\$:	CMP	PRSTRK,LSTTRK	;ARE WE DONE?
2829	031414	001001			BNE	5\$;NO, KEEP LOOKING

GLOBAL SUBROUTINES SECTION

```

2830 031416 000205          RTS      R5          ;EXIT WITH NOT FOUND
2831 031420 063737 003146 003154 5$:  ADD      OFFSET,PRSTRK ;NEXT TRACK
2832 031426 000740          BR       1$
2833
2834          ;ROUTINE TO FIND BAD TRACK IN FILE
2835          ;CALL   JSR      R5,FNDBSC
2836
2837 031430 005037 002234  FNDBSC: CLR      HDRFND          ;INITIALIZE FLAG
2838 031434 010146          MOV      R1,-(SP)        ;SAVE R1, R2
2839 031436 010246          MOV      R2,-(SP)
2840 031440 012701 003242          MOV      @BUF+10,R1      ;SETUP FOR BEGINNING OF FILE
2841 031444 005711          1$:   TST      (R1)          ;END?
2842 031446 100421          BMI      2$             ;IF MINUS AT END, EXIT
2843 031450 023721 003154          CMP      PRSTRK,(R1)+   ;CYLINDER CORRECT?
2844 031454 001011          BNE      3$             ;NO, NEXT
2845 031456 105724          TSTB    (R4)+          ;UPPER HALF OF WORD
2846 031460 123711 003156          CMPB    SURFACE,(R1)   ;CORRECT SURFACT
2847 031464 001402          BEQ     4$
2848 031466 105744          TSTB    -(R4)
2849 031470 000403          BR      3$
2850 031472 005237 002234          4$:   INC      HDRFND          ;SET FOUND
2851 031476 000405          BR      2$
2852
2853 031500 005721          3$:   TST      (R1)+          ;NEXT WORD
2854 031502 005202          INC      R2             ;ACCOUNT FOR IT
2855 031504 020227 000374          CMP      R2,@252.      ;DONE?
2856 031510 001355          BNE      1$             ;NO, KEEP CHECKING
2857 031512 012601          2$:   MOV      (SP)+,R1      ;RESTORE R2, R1, EXIT
2858 031514 012602          MOV      (SP)+,R2
2859 031516 000205          RTS      R5
2860
2861 031520 013701 003154  FIXCYL: MOV      PRSTRK,R1      ;GET TRACK WHICH IS GOOD
2862 031524 005737 003146          TST      OFFSET        ;WHICH WAY WERE WE LOOKING
2863 031530 100402          BMI      1$             ;IN WORD, BRANCH
2864 031532 162701 000004          SUB      @4,R1          ;BACK IT UP BY FOUR
2865 031536 012702 000005          1$:   MOV      @5,R2          ;GOING STORE AWAY 5 TRACKS
2866 031542 010120          2$:   MOV      R1,(R0)+     ;STORE THEM 1 WD/PER
2867 031544 005201          INC      R1
2868 031546 005302          DEC      R2
2869 031550 001374          BNE      2$
2870 031552 000205          RTS      R5
2871
2872          ;ROUTINE TO GET SERIAL NUMBER
2873
2874          ;CALL   JSR      R5,SERNUM
2875
2876 031554 012737 000013 003204  SERNUM: MOV      @13,BDA
2877 031562 004537 032432          JSR      R5,LDFUNC      ;GET STATUS
2878 031566 000004          GSTAT
2879 031570 004537 032432          JSR      R5,LDFUNC      ;READ HEADER
2880 031574 000010          RDHDR
2881 031576 013700 003172          MOV      E.MP,R0        ;GET THE HEADER
2882 031602 042700 000077          1$:   BIC      @77,R0          ;CLEAR SECTOR BITS
2883 031606 022737 000001 003060          CMP      @1,T.DRIVE
2884 031614 001003          BNE      23$
2885 031616 020027 077700          CMP      R0,@77700
2886 031622 001446          BEQ     2$

```


GLOBAL SUBROUTINES SECTION

```

2887 031624 020027 177700      23$:  CMP      R0,#177700
2888 031630 001443              BEQ      2$
2889 031632 042700 000100      BIC      #100,R0      ;CLEAR HEAD
2890 031636 022737 000001 003060  CMP      #1,T.DRIVE
2891 031644 001003              BNE      32$
2892 031646 012701 077600      MOV      #77600,R1
2893 031652 000402              BR       33$
2894 031654 012701 177600      32$:  MOV      #177600,R1
2895
2896 031660 160001              33$:  SUB      R0,R1
2897 031662 010137 003204      MOV      R1,BDA      ;SET UP DIF WORD
2898 031666 052737 000025 003204  BIS      #25,BDA      ;SEEK IN, HEAD 1
2899 031674 004537 032432      JSR      R5,LDFUNC    ;SEEK
2900 031700 000006              SEEK
2901 031702 004537 032432      JSR      R5,LDFUNC    ;VERIFY POSITION
2902 031706 000010      RDHDR
2903 031710 013700 003172      MOV      E.MP,R0      ;GET HEADER
2904 031714 022737 000001 003060  CMP      #1,T.DRIVE
2905 031722 001003              BNE      42$
2906 031724 022700 077700      CMP      #77700,R0
2907 031730 000402              BR       43$
2908 031732 022700 177700      42$:  CMP      #177700,R0
2909
2910 031736 103321              43$:  BHIS     1$
2911 031740 022737 000001 003060  2$:  CMP      #1,T.DRIVE
2912 031746 001004              BNE      52$
2913 031750 012737 077700 003204  MOV      #77700,BDA
2914 031756 000403              BR       97$
2915
2916 031760 012737 177700 003204  52$:  MOV      #177700,BDA
2917 031766 012737 003232 003202  97$:  MOV      #BUF,BBA
2918 031774 012737 177400 003206  MOV      #-256.,BMP
2919 032002 004537 032432      JSR      R5,LDFUNC    ;READ
2920 032006 000014      READ
2921 032010 005737 003074      TST      ERFLG      ;TEST ERROR FLAG
2922 032014 001421              BEQ      98$          ;YES,COMPARE SERIAL NUMBERS
2923 032016 062737 000004 003204  ADD      #4,BDA      ;NO,SETUP FOR NEXT FACTORY BAD SECTOR
2924 032024 022737 000001 003060  CMP      #1,T.DRIVE
2925 032032 001005              BNE      62$
2926 032034 022737 077724 003204  CMP      #77724,BDA
2927 032042 001351              BNE      97$
2928 032044 000453              BR       99$
2929 032046 022737 177724 003204  62$:  CMP      #177724,BDA
2930 032054 001344              BNE      97$
2931 032056 000446              BR       99$          ;GET NEXT FACTORY BAD SECTOR
2932 032060 012701 003232      98$:  MOV      #BUF,R1      ;REPORT ERROR
2933 032064 005737 003210      TST      SERNM1      ;COMPARE SERIAL NUMBERS
2934 032070 100005              BPL      3$          ;HAVE WE GOT ONE TO COMPARE
2935 032072 011137 003210      MOV      (R1),SERNM1 ;YES, BRANCH
2936 032076 016137 000002 003212  MOV      2(R1),SERNM2 ;NO, CALL THIS ONE IT
2937 032104 021137 003210      3$:  CMP      (R1),SERNM1 ;SERNUM OKAY
2938 032110 001004              BNE      4$          ;NO, PRINT ERROR
2939 032112 026137 000002 003212  CMP      2(R1),SERNM2 ;OTHER HALF OKAY
2940 032120 001437              BEQ      5$          ;YES, EXIT
2941 032122      4$:  PRINTF   #FRM3,2(R1),(R1),SERNM2,SERNM1
      032122 013746 003210      MOV      SERNM1,-(SP)
      032126 013746 003212      MOV      SERNM2,-(SP)

```

GLOBAL SUBROUTINES SECTION

```

032132 011146          MOV      (R1), -(SP)
032134 016146 000002  MOV      2(R1), -(SP)
032140 012746 021167  MOV      #FRM3, -(SP)
032144 012746 000005  MOV      #5, -(SP)
032150 010600          MOV      SP, R0
032152 104417          TRAP     C:PNTF
032154 062706 000014  ADD      #14, SP
2942 032160 004537 032222  JSR      R5, UNLOAD      ;LET OPERATOR CHANGE
2943 032164 004537 032326  JSR      R5, LOAD        ;PACK
2944 032170 000137 031554  JMP      SERNUM          ;GO CHECK IT AGAIN.
2945 032174          99$: PRINTF  #FRM15          ;MESSAGE
032174 012746 022037  MOV      #FRM15, -(SP)
032200 012746 000001  MOV      #1, -(SP)
032204 010600          MOV      SP, R0
032206 104417          TRAP     C:PNTF
032210 062706 000004  ADD      #4, SP
2946 032214          999$: BREAK
032214 104422          TRAP     C:BRK
2947 032216 000776          BR       999$
2948 032220 000205          5$:   RTS      R5

;ROUTINE UNLOAD
;CALL JSR      R5, UNLOAD

UNLOAD: PRINTF  #FRM1, <B, DSB+1(R4)>, CSR(R4) ;PROMPT - UNLOAD DRIVE - ON CONTROLLER -
;AND REMOVE PACK
MOV      CSR(R4), -(SP)
CLR      -(SP)
BISB    DSB+1(R4), (SP)
MOV      #FRM1, -(SP)
MOV      #3, -(SP)
MOV      SP, R0
TRAP     C:PNTF
ADD      #10, SP
MOV      #60, R1          ;SETUP 60 SECOND TIMER
MOV      #200, R0
BIS     DSB(R4), R0
MOV      R0, BCSR(R4)
2959 032274 032774 000001 000000 2$:   BIT      #DRDY, BCSR(R4) ;CHECK DRDY FOR ZERO
2960 032302 001410          BEQ     3$              ;PACK UNLOADED
2961 032304          WAITMS #10.          ;WAIT 1 SECOND
2962 032316 005301          DEC     R1             ;HAS 60 SEC PASSED?
2963 032320 001365          BNE     2$              ;NO, RETEST DRDY, CONTINUE WAIT
2964 032322 000737          BR      UNLOAD         ;YES, REPEAT MESSAGE CONTINUE WAIT
2965 032324 000205          3$:   RTS      R5              ;RETURN WITH PACK UNLOADED

;ROUTINE LOAD
;CALL JSR      R5, LOAD

LOAD:   PRINTF  #FRM2, <B, DSB+1(R4)>, CSR(R4) ;PLACE PACK IN DRIVE - ON CONTROLLER - AND
;LOAD IT
MOV      CSR(R4), -(SP)
CLR      -(SP)
BISB    DSB+1(R4), (SP)
MOV      #FRM2, -(SP)
MOV      #3, -(SP)
MOV      SP, R0
    
```

GLOBAL SUBROUTINES SECTION

```

032352 104417 TRAP C:PNTF
032354 062706 000010 ADD #10,SP
2972 032360 012701 000170 MOV #120.,R1 ;SETUP 120 SEC TIMER
2973 032364 012700 000200 MOV #200,R0 ;SETUP CONTROLLER READY BIT
2974 032370 056400 000004 BIS DSB(R4),R0 ;SELECT DRIVE
2975 032374 010074 000000 MOV R0,@CSR(R4)
2976 032400 032774 000001 000000 2#: BIT #DRDY,@CSR(R4)
2977 032406 001010 BNE 3#
2978 032410 WAITMS #10.
2979 032422 005301 DEC R1
2980 032424 001365 BNE 2#
2981 032426 000737 BR LOAD
2982
2983 032430 000205 3#: RTS R5
2984
2985 ;ROUTINE LDFUNC
2986 ;CALL JSR R5,LDFUNC
2987
LDFUNC: MOV R0,-(SP)
2988 032432 010046 MOV R3,-(SP)
2989 032434 010346 MOV R1,-(SP)
2990 032436 010146 CLR ERFLG ;CLEAR ERROR FLAG
2991 032440 005037 003074 MOV CSR(R4),R3 ;GET CSR
2992 032444 016403 000000 MOV BMP,MP(R3) ;LOAD MULTIPURPOSE
2993 032450 013763 003206 000006 MOV BDA,DA(R3) ;LOAD DISK ADDRESS
2994 032456 013763 003204 000004 MOV BBA,BA(R3) ;LOAD BUS ADDRESS
2995 032464 013763 003202 000002 MOV (R5),BCS ;GET FUNCTION TO LOAD
2996 032472 011537 003200 BIS DSB(R4),BCS ;SELECT BITS
2997 032476 056437 000004 003200 MOV #25.,R1 ;SET WATCHDOG TO 250MS
2998 032504 012701 000031 BIS #200,BCS
2999 032510 052737 000200 003200 MOV BCS,CS(R3) ;LOAD FUNCTION
3000 032516 013763 003200 000000 MOV CS(R3),BCS
3001 032524 016337 000000 003200 BIC #200,CS(R3)
3002 032532 042763 000200 000000 1#: BIT #200,CS(R3) ;CNTLR READY?
3003 032540 032763 000200 000000 BNE 2# ;YES, GO
3004 032546 001036 WAITUS #100. ;WAIT 10 MILLISECONDS
3005 032550 DEC R1
3006 032562 005301 BNE 1#
3007 032564 001365
3008
3009 032566 016337 000000 003164 MOV CS(R3),E.CS ;READ ALL REGISTERS
3010 032574 016337 000002 003166 MOV BA(R3),E.BA
3011 032602 016337 000004 003170 MOV DA(R3),E.DA
3012 032610 016337 000006 003172 MOV MP(R3),E.MP
3013 032616 016337 000006 003174 MOV MP(R3),E.MP1
3014 032624 016337 000006 003176 MOV MP(R3),E.MP2
3015 032632 ERDF 210.,CNTTOT,ERR5;CNTLR TIMEOUT
032632 104455 TRAP C:ERDF
032634 000322 .WORD 210
032636 017346 .WORD CNTTOT
032640 020644 .WORD ERR5
3016 032642 000425 BR 4#
3017
3018 032644 016337 000000 003164 2#: MOV CS(R3),E.CS ;READ ALL REGISTERS
3019 032652 016337 000002 003166 MOV BA(R3),E.BA
3020 032660 016337 000004 003170 MOV DA(R3),E.DA
3021 032664 016337 000006 003172 MOV MP(R3),E.MP
3022 032668 016337 000006 003174 MOV MP(R3),E.MP1

```

GLOBAL SUBROUTINES SECTION

```

3023 032702 016337 000006 003176      MOV      MP(R3),E,MP2
3024
3025 032710 005737 003164      TST      E.CS          ;ANY ERRORS?
3026 032714 100002          BPL      3$            ;YES, GO SERVICE
3027 032716 005237 003074      4$:     INC      ERFLG
3028 032722 005725          3$:     TST      (R5)+
3029 032724 012601      MOV      (SP)+,R1
3030 032726 012603      MOV      (SP)+,R3
3031 032730 012600      MOV      (SP)+,R0
3032 032732 000205      RTS      R5
3033
3034 032734          ENDMOD
3035
3036 032734          BGNMOD  HRDWTST
3037
3038          .SBTTL CONTROL ROUTINE
3039
3040 032734          BGNTST
3041
3042          ;CONTROL SECTION COMPATIBILITY PROGRAM
3043          ;PRINT UNLOAD AND LOAD DRIVE MESSAGES
3044          ;PERFORM SERIAL CHECK ROUTINE
3045          ;PERFORM READ/WRITE CHECKS ON DRIVES
3046
3047 032734 012701 002442      COMPAT: MOV      @SECBUF,R1      ;ADJ. CYLINDER BUFFER
3048 032740 012700 000170      MOV      @120.,R0          ;ADJ. CYLINDER BUFFER COUNT
3049 032744 005021          4$:     CLR      (R1)+        ;CLEAR ADJ. CYL. BUFFER AT STARTUP
3050 032746 005300          DEC      R0                ;BUFFER CLEARED?
3051 032750 001375          BNE      4$                ;CLEAR NEXT BUFFER WORD
3052 032752 005237 003062      INC      F0WR              ;SET FIRST OVERWRITE FLAG
3053 032756 004537 024634      JSR      R5,OVWPER          ;PERFORM OVERWRITE ON FIRST DRIVE
3054 032762 177400          177400
3055 032764 000377          377
3056 032766 005037 003062      CLR      F0WR              ;CLEAR FIRST OVERWRITE
3057 032772 005237 003064      INC      FADJ              ;SET FIRST ADJ. FLAG
3058 032776 005237 003104      INC      ADJDIR            ;UP = 1
3059 033002 004537 025356      JSR      R5,ADJCYL
3060 033006 003 377          .BYTE   3,377              ;TRACK AND SECTORS FOR
3061 033010 170000          .WORD   170000            ;INWARD SEEK
3062 033012 003 000          .BYTE   3,0                ;TRACK AND SECTORS FOR
3063 033014 007777          .WORD   7777              ;OUTWARD SEEK
3064 033016 000000          .WORD   0                  ;TERMINATOR
3065 033020 004537 032222      JSR      R5,UNLOAD          ;UNLOAD PACK FROM DRIVE UNIT
3066 033024 062704 000010      ADD      @PAT+2,R4          ;UPDATE POINTER FOR NEXT DRIVE
3067 033030 004537 032326      JSR      R5,LOAD            ;LOAD INTO SECOND DRIVE UNIT
3068 033034 004537 031554      JSR      R5,SERNUM          ;CHECK PACK SERIAL NUMBER
3069 033040 004537 024634      JSR      R5,OVWPER          ;PERFORM R/W OVERWRITE
3070 033044 000360          360
3071 033046 000017          17
3072 033050 005237 003104      INC      ADJDIR
3073 033054 004537 025356      JSR      R5,ADJCYL
3074 033060 002 360          .BYTE   2,360              ;IN 1/0 OUTSIDE
3075 033062 000000          .WORD   0
3076 033064 002 017          .BYTE   2,17              ;OUT 1/0 OUTSIDE
3077 033066 000000          .WORD   0
3078 033070 004 360          .BYTE   4,360              ;IN 1/0 INSIDE
3079 033072 000000          .WORD   0

```

CONTROL ROUTINE

3080	033074	004	017		.BYTE	4,17		;OUT 1/0 INSIDE
3081	033076	000000			.WORD	0		
3082	033100	000000			.WORD	0		
3083	033102	004537	032222		JSR	R5,UNLOAD		;UNLOAD PACK FROM DRIVE UNIT
3084	033106	023727	003130	000002	CMP	UUT.#2		;CHECK FOR > 2 DRIVES
3085	033114	001002			BNE	10#		;YES,GO TO NEXT DRIVE
3086	033116	000137	033532		JMP	2#		;GO TO FIRST DRIVE
3087	033122	062704	000010	10#:	ADD	@PAT+2,R4		;UPDATE DRIVE BUFFER FOR THIRD DRIVE
3088	033126	004537	032326		JSR	R5,LOAD		;LOAD PACK FOR THIRD DRIVE
3089	033132	004537	031554		JSR	R5,SERNUM		;CHECK SERIAL NUMBERS
3090	033136	004537	024634		JSR	R5,OVWPER		;PERFORM R/W OVERWRITE ON THIRD DRIVE
3091	033142	006014				6014		
3092	033144	001403				1403		
3093	033146	005237	003104		INC	ADJDIR		
3094	033152	004537	025356		JSR	R5,ADJCYL		
3095	033156	002	000		.BYTE	2,0		;IN 2/0 OUTSIDE
3096	033160	170000			.WORD	170000		
3097	033162	002	000		.BYTE	2,0		;OUT 2/0 OUTSIDE
3098	033164	007400			.WORD	7400		
3099	033166	004	000		.BYTE	4,0		;IN 2/0 INSIDE
3100	033170	170000			.WORD	170000		
3101	033172	004	000		.BYTE	4,0		;OUT 2/0 INSIDE
3102	033174	007400			.WORD	7400		
3103	033176	001	200		.BYTE	1,200		;IN 2/1 OUTSIDE
3104	033200	000000			.WORD	0		
3105	033202	001	100		.BYTE	1,100		;OUT 2/1 OUTSIDE
3106	033204	000000			.WORD	0		
3107	033206	005	200		.BYTE	5,200		;IN 2/1 INSIDE
3108	033210	000000			.WORD	0		
3109	033212	005	100		.BYTE	5,100		;OUT 2/1 INSIDE
3110	033214	000000			.WORD	0		
3111	033216	000000			.WORD	0		;TERMINATOR
3112	033220	004537	032222		JSR	R5,UNLOAD		;UNLOAD PACK ON THIRD DRIVE
3113	033224	023727	003130	000003	CMP	UUT,#3		;CHECK FOR > 3 DRIVES
3114	033232	001500			BEG	1#		;NO, GO TO 2ND DRIVE
3115	033234	062704	000010		ADD	@PAT+2,R4		;UPDATE DRIVE BUFFER FOR 4TH DRIVE
3116	033240	004537	032326		JSR	R5,LOAD		;LOAD PACK ON 4TH DRIVE
3117	033244	004537	031554		JSR	R5,SERNUM		;CHECK PACK ON FOURTH DRIVE
3118	033250	004537	024634		JSR	R5,OVWPER		;PERFORM R/W OVERWRITE
3119	033254	001042				1042		
3120	033256	000421				421		
3121	033260	005237	003104		INC	ADJDIR		
3122	033264	004537	025356		JSR	R5,ADJCYL		
3123	033270	002	000		.BYTE	2,0		;IN 3/0 OUTSIDE
3124	033272	000360			.WORD	360		
3125	033274	002	000		.BYTE	2,0		;OUT 3/0 OUTSIDE
3126	033276	000017			.WORD	17		
3127	033300	004	000		.BYTE	4,0		;IN 3/0 INSIDE
3128	033302	000360			.WORD	360		
3129	033304	004	000		.BYTE	4,0		;OUT 3/0 INSIDE
3130	033306	000017			.WORD	17		
3131	033310	001	040		.BYTE	1,40		;IN 3/1 OUTSIDE
3132	033312	000000			.WORD	0		
3133	033314	001	020		.BYTE	1,20		;OUT 3/1 OUTSIDE
3134	033316	000000			.WORD	0		
3135	033320	005	040		.BYTE	5,40		;IN 3/1 INSIDE
3136	033322	000000			.WORD	0		

CONTROL ROUTINE

3137	033324	005	020	.BYTE	5,20	;OUT 3/1 INSIDE
3138	033326	000000		.WORD	0	
3139	033330	001	000	.BYTE	1,0	;IN 3/2 OUTSIDE
3140	033332	100000		.WORD	100000	
3141	033334	001	000	.BYTE	1,0	;OUT 3/2 OUTSIDE
3142	033336	040000		.WORD	40000	
3143	033340	005	000	.BYTE	5,0	;IN 3/2 INSIDE
3144	033342	100000		.WORD	100000	
3145	033344	005	000	.BYTE	5,0	;OUT 3/2 INSIDE
3146	033346	040000		.WORD	40000	
3147	033350	000000		.WORD	0	; TERMINATOR
3148	033352	004537	032222	JSR	R5,UNLOAD	;UNLOAD PACK FROM 4TH DRIVE
3149	033356	162704	000010	SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 3RD DRIVE
3150	033362	004537	032326	JSR	R5,LOAD	;LOAD PACK ON 3RD DRIVE
3151	033366	004537	031554	JSR	R5,SERNUM	;CHECK FOR PACK SERIAL NUMBER
3152	033372	004537	024634	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE ON 3RD DRIVE
3153	033376	020000		.WORD	20000	
3154	033400	010000		.WORD	10000	
3155	033402	004537	025356	JSR	R5,ADJCYL	
3156	033406	001	000	.BYTE	1,0	;IN 2/3 OUTSIDE
3157	033410	000200		.WORD	200	
3158	033412	001	000	.BYTE	1,0	;OUT 2/3 OUTSIDE
3159	033414	000100		.WORD	100	
3160	033416	005	000	.BYTE	5,0	;IN 2/3 INSIDE
3161	033420	000200		.WORD	200	
3162	033422	005	000	.BYTE	5,0	;OUT 2/3 INSIDE
3163	033424	000100		.WORD	100	
3164	033426	000000		.WORD	0	; TERMINATOR
3165	033430	004537	032222	JSR	R5,UNLOAD	;UNLOAD PACK FROM 3RD DRIVE
3166	033434	162704	000010	SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 2ND DRIVE
3167	033440	004537	032326	JSR	R5,LOAD	;LOAD PACK ON THIRD DRIVE
3168	033444	004537	031554	JSR	R5,SERNUM	;CHECK PACK SERIAL NUMBER
3169	033450	004537	024634	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE ON 2ND DRIVE
3170	033454	004040		.WORD	4040	
3171	033456	002020		.WORD	2020	
3172	033460	004537	025356	JSR	R5,ADJCYL	
3173	033464	001	000	.BYTE	1,0	;IN 1/2 OUTSIDE
3174	033466	020000		.WORD	20000	
3175	033470	001	000	.BYTE	1,0	;OUT 1/2 OUTSIDE
3176	033472	010000		.WORD	10000	
3177	033474	005	000	.BYTE	5,0	;IN 1/2 INSIDE
3178	033476	020000		.WORD	20000	
3179	033500	005	000	.BYTE	5,0	;OUT 1/2 INSIDE
3180	033502	010000		.WORD	10000	
3181	033504	001	000	.BYTE	1,0	;IN 1/3 OUTSIDE
3182	033506	000040		.WORD	40	
3183	033510	001	000	.BYTE	1,0	;OUT 1/3 OUTSIDE
3184	033512	000020		.WORD	20	
3185	033514	005	000	.BYTE	5,0	;IN 1/3 INSIDE
3186	033516	000040		.WORD	40	
3187	033520	005	000	.BYTE	5,0	;OUT 1/3 INSIDE
3188	033522	000020		.WORD	20	
3189	033524	000000		.WORD	0	; TERMINATOR
3190	033526	004537	032222	JSR	R5,UNLOAD	;UNLOAD PACK FROM 2ND DRIVE
3191	033532	162704	000010	SUB	#PAT+2,R4	;SET DRIVE BUFFER FOR 1ST DRIVE
3192	033536	004537	032326	JSR	R5,LOAD	;LOAD PACK INTO FIRST DRIVE UNIT
3193	033542	004537	031554	JSR	R5,SERNUM	;CHECK SERIAL NUMBER

14:

24:

CONTROL ROUTINE

3194	033546	004537	024634	JSR	R5,OVWPER	;PERFORM R/W OVERWRITE
3195	033552	001042		1042		
3196	033554	000421		421		
3197	033556	004537	025356	JSR	R5,ADJCYL	
3198	033562	001	010	.BYTE	1,10	;IN 0/1 OUTSIDE
3199	033564	000000		.WORD	0	
3200	033566	001	004	.BYTE	1,4	;OUT 0/1 OUTSIDE
3201	033570	000000		.WORD	0	
3202	033572	005	010	.BYTE	5,10	;IN 0/1 INSIDE
3203	033574	000000		.WORD	0	
3204	033576	005	004	.BYTE	5,4	;OUT 0/1 INSIDE
3205	033600	000000		.WORD	0	
3206	033602	001	000	.BYTE	1,0	;IN 0/2 OUTSIDE
3207	033604	004000		.WORD	4000	
3208	033606	001	000	.BYTE	1,0	;OUT 0/2 OUTSIDE
3209	033610	002000		.WORD	2000	
3210	033612	005	000	.BYTE	5,0	;IN 0/2 INSIDE
3211	033614	004000		.WORD	4000	
3212	033616	005	000	.BYTE	5,0	;OUT 0/2 INSIDE
3213	033620	002000		.WORD	2000	
3214	033622	001	000	.BYTE	1,0	;IN 0/3 OUTSIDE
3215	033624	000010		.WORD	10	
3216	033626	001	000	.BYTE	1,0	;OUT 0/3 OUTSIDE
3217	033630	000004		.WORD	4	
3218	033632	005	000	.BYTE	5,0	;IN 0/3 INSIDE
3219	033634	000010		.WORD	10	
3220	033636	005	000	.BYTE	5,0	;OUT 0/3 INSIDE
3221	033640	000004		.WORD	4	
3222	033642	000000		.WORD	0	;TERMINATOR
3223	033644	004537	032222	JSR	R5,UNLOAD	;UNLOAD PACK
3224	033650			PRINTF	#ENDPAS	;END OF PASS
	033650	012746	022412	MOV	#ENDPAS,-(SP)	
	033654	012746	000001	MOV	#1,-(SP)	
	033660	010600		MOV	SP,R0	
	033662	104417		TRAP	C#PNTF	
	033664	062706	000004	ADD	#4,SP	
3225						
3226	033670	000137	024314	JMP	CHPENA	;RETURN TO SUPERVISOR
3227						
3228	033674			ENDTST		
	033674			L10014:		
	033674	104401		TRAP	C#ETST	
3229	033676			ENDMOD		
3230						
3231	033676			BGNMOD	HRDPRM	
3232	033676			BGNHRD		
	033676	000025		.WORD	L10015-L#HARD/2	
3233						
3234	033700			GPRMA	CSRMSG,CSR,0,160000,177776,YES	
	033700	000031		.WORD	T#CODE	
	033702	033752		.WORD	CSRMSG	
	033704	160000		.WORD	T#LOLIM	
	033706	177776		.WORD	T#HILIM	
3235						
3236	033710			GPRMA	VECMMSG,VECT,0,0,776,YES	
	033710	001031		.WORD	T#CODE	
	033712	034010		.WORD	VECMMSG	

CONTROL ROUTINE

```

033714 000000 .WORD T$LOLIM
033716 000776 .WORD T$HILIM
3237
3238 033720 GPRMD DRMSG,DRBT,0,03400,0,7,YES
033720 004032 .WORD T$CODE
033722 034017 .WORD DRMSG
033724 003400 .WORD 03400
033726 000000 .WORD T$LOLIM
033730 000007 .WORD T$HILIM
3239
3240 033732 GPRML DRTYPE,TYPDR,1,YES
033732 003130 .WORD T$CODE
033734 033766 .WORD DRTYPE
033736 000001 .WORD 1
3241
3242 033740 GPRMD BRMSG,PRIOR,0,340,0,7,YES
033740 002032 .WORD T$CODE
033742 034025 .WORD BRMSG
033744 000340 .WORD 340
033746 000000 .WORD T$LOLIM
033750 000007 .WORD T$HILIM
3243
3244 033752 ENDHRD
033752 .EVEN
L10015:
3245
3246 033752 102 125 123 CSRMSG: .ASCIZ /BUS ADDRESS/
033755 040 101 104
033760 104 122 105
033763 123 123 000
3247 033766 104 122 111 DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
033771 126 105 040
033774 124 131 120
033777 105 040 075
034002 040 122 114
034005 060 061 000
3248 034010 126 105 103 VECMSG: .ASCIZ /VECTOR/
034013 124 117 122
034016 000
3249 034017 104 122 111 DRMSG: .ASCIZ /DRIVE/
034022 126 105 000
3250 034025 102 122 040 BRMSG: .ASCIZ /BR LEVEL/
034030 114 105 126
034033 105 114 000
3251
3252 .EVEN
3253
3254 034036 ENDMOD
3255
3256 034036 LASTAD
034036 000000 .EVEN
034040 000000 .WORD 0
034042 .WORD 0
L$LAST::
3257
3258 000001 .END

```


SYMBOL TABLE

ADJCYL	025356	CRDY	= 000200	C#SVEC	= 000037	FRM12	021646	G#OF SI	= 000376
ADJDIR	003104	CRSET	= 000002	C#TPRI	= 000013	FRM13	021725	G#PRMA	= 000001
ADJFLG	003102	CS	= 000000	DA	= 000004	FRM14	022010	G#PRMD	= 000002
ADJLC2	003220	CSR	= 000000	DCKER	017455	FRM15	022037	G#PRML	= 000000
ADJLC3	003222	CSRMSG	033752	DCRC	= 004000	FRM16	022076	G#RADA	= 000140
ADJLC4	003224	CYL	003122	DERR	= 040000	FRM17	022163	G#RADB	= 000000
ADJLOC	003100	C#AU	= 000052	DIAGMC	= 000000	FRM18	022217	G#RADD	= 000040
ADJTRK	003214	C#AUTO	= 000061	DIRC	003116	FRM19	022303	G#RADL	= 000120
ADJTXT	020210	C#BRK	= 000022	DLT	= 010000	FRM2	021067	G#RADO	= 000020
ADJUUT	003216	C#BSEG	= 000004	DRBT	= 000010	FRM20	022341	G#XFER	= 000004
ADR	= 000020	C#BSUB	= 000002	DRBUF	017240	FRM3	021167	G#YES	= 000010
ASSEMB	= 000010	C#CEFG	= 000045	DRDY	= 000001	FRM4	021246	HCRC	= 004000
AUTOCO	024340	C#CLCK	= 000062	DRMSG	034017	FRM5	021307	HDRFND	002234
BA	= 000002	C#CLEA	= 000012	DRPCOD	024350	FRM6	021356	HEAD	= 000100
BA16	= 000020	C#CLOS	= 000035	DRST	= 000013	FRM7	021377	HEAD01	003114
BA17	= 000040	C#CLP1	= 000006	DRSTAT	003106	FRM8	021437	HNF	= 010000
BBA	003202	C#CVEC	= 000036	DRTYPE	033766	FRM9	021503	HOE	= 100000
BCS	003200	C#DCLN	= 000044	DSB	= 000004	FRTTRK	003152	HPTCOD	022446
BDA	003204	C#DODU	= 000051	DSPCOD	022462	FUNERR	020047	HRDPRM	033676
BDATA	003140	C#DRPT	= 000024	EF.CON	= 000036	FWD	020116	HRDWS	032734
BIT0	= 000001	C#DU	= 000053	EF.NEW	= 000035	F#AU	= 000015	HSFLG	003110
BIT00	= 000001	C#EDIT	= 000003	EF.PMR	= 000034	F#AUTO	= 000020	IBE	= 010000
BIT01	= 000002	C#ERDF	= 000055	EF.RES	= 000037	F#BGN	= 000040	IDU	= 000040
BIT02	= 000004	C#ERHR	= 000056	EF.STA	= 000040	F#CLEA	= 000007	IER	= 020000
BIT03	= 000010	C#ERRO	= 000060	END	023046	F#DU	= 000016	INAWR	030502
BIT04	= 000020	C#ERSF	= 000054	ENDBUF	017300	F#END	= 000041	INITCO	022466
BIT05	= 000040	C#ERSO	= 000057	ENDPAS	022412	F#HARD	= 000004	INITWR	017373
BIT06	= 000100	C#ESCA	= 000010	ERFLG	003074	F#HW	= 000013	INN10	002356
BIT07	= 000200	C#ESEG	= 000005	ERR	= 100000	F#INIT	= 000006	INN11	002370
BIT08	= 000400	C#ESUB	= 000003	ERRFND	017743	F#JMP	= 000050	INN20	002360
BIT09	= 001000	C#ETST	= 000001	ERR1	020240	F#MOD	= 000000	INN21	002372
BIT1	= 000002	C#EXIT	= 000032	ERR2	020276	F#MSG	= 000011	INN30	002362
BIT10	= 002000	C#GETB	= 000026	ERR3	020336	F#PROT	= 000021	INN31	002374
BIT11	= 004000	C#GETW	= 000027	ERR4	020464	F#PWR	= 000017	INN40	002364
BIT12	= 010000	C#GMAN	= 000043	ERR5	020644	F#RPT	= 000012	INN41	002376
BIT13	= 020000	C#GPMR	= 000042	ERR6	020704	F#SEG	= 000003	INN50	002366
BIT14	= 040000	C#GPLO	= 000030	EVL	= 000004	F#SOFT	= 000005	INN51	002400
BIT15	= 100000	C#GPRI	= 000040	EXIT	024336	F#SRV	= 000010	INTEN	= 000100
BIT2	= 000004	C#INIT	= 000011	E#END	= 002100	F#SUB	= 000002	ISR	= 000100
BIT3	= 000010	C#INLP	= 000020	E#LOAD	= 000035	F#SW	= 000014	IXE	= 004000
BIT4	= 000020	C#MANI	= 000050	E.BA	003166	F#TEST	= 000001	I#AU	= 000041
BIT5	= 000040	C#MEH	= 000031	E.CS	003164	GDATA	003136	I#AUTO	= 000041
BIT6	= 000100	C#MSG	= 000023	E.DA	003170	GLBDAT	002234	I#CLN	= 000041
BIT7	= 000200	C#OPEN	= 000034	E.MP	003172	GLBEQA	002234	I#DU	= 000041
BIT8	= 000400	C#PNTB	= 000014	E.MP1	003174	GLBERR	020240	I#HRD	= 000041
BIT9	= 001000	C#PNTF	= 000017	E.MP2	003176	GLBSUB	024354	I#INIT	= 000041
BMP	003206	C#PNTS	= 000016	FADJ	003064	GLBTXT	017302	I#MOD	= 000041
BOE	= 000400	C#PNTX	= 000015	FEW	017473	G\$BIT	= 000003	I#MSG	= 000041
BRMSG	034025	C#QIO	= 000377	FIXCYL	031520	G\$STAT	= 000004	I#PROT	= 000040
BSEXIT	030566	C#RDBU	= 000007	FND\$BSC	031430	G#CNT0	= 000200	I#PTAB	= 000041
BSVWR	030404	C#REFG	= 000047	FND\$DRV	027166	G#DELM	= 000372	I#PWR	= 000041
BUF	003232	C#RESE	= 000033	FND\$TRK	031256	G#DISP	= 000003	I#RPT	= 000041
CHECK	030614	C#REVI	= 000003	FORSK	003126	G#EXCP	= 000400	I#SEG	= 000041
CLNCOD	024344	C#RFLA	= 000021	FOWR	003062	G#HILI	= 000002	I#SETU	= 000041
COMPENA	024314	C#RPT	= 000025	FRM1	020772	G#LOLI	= 000001	I#SRV	= 000041
CNTTOT	017346	C#SEFG	= 000046	FRM10	021545	G#NO	= 000000	I#SUB	= 000041
COMPAT	032734	C#SPRI	= 000041	FRM11	021611	G#OFFS	= 000400	I#TST	= 000041

SYMBOL TABLE

J\$JMP = 000167	L\$SPC 002056 G	OQU40 002270	REV 020126	T\$HILI= 000007
LDFUNC 032432	L\$SPCP 002020 G	OQU41 002302	REVSJ 003124	T\$LAST= 000001
LOAD 032326	L\$SPTP 002024 G	OQU50 002272	RSADJS 026760	T\$LOLI= 000000
LOE = 040000 G	L\$STA 002030 G	OQU51 002304	SECBUF 002442	T\$LSYM= 010000
LOT = 000010 G	L\$TEST 002114 G	OSECT 003112	SECLST 002402	T\$LTNO= 000001
LSTCLR 003070	L\$TIML 002014 G	OUT10 002236	SECT 003132	T\$NEST= 177777
LSTDRV 003134	L\$UNIT 002012 G	OUT11 002250	SECWRD 003144	T\$NSO = 000000
LSTTRK 003150	L10000 020274	OUT20 002240	SEEK = 000006	T\$NS1 = 000004
L\$ACP 002110 G	L10001 020334	OUT21 002252	SERNM1 003210	T\$PTNU= 000000
L\$APT 002036 G	L10002 020462	OUT30 002242	SERNM2 003212	T\$SAVL= 177777
L\$AUT 002070 G	L10003 020642	OUT31 002254	SERNUM 031554	T\$SEGL= 177777
L\$AUTO 024340 G	L10004 020702	OUT40 002244	SETLST 027126	T\$SUBN= 000000
L\$CCP 002106 G	L10005 020770	OUT41 002256	SETUP 023120	T\$TAGL= 177777
L\$CLEA 024344 G	L10007 022462	OUT50 002246	SIGN = 000004	T\$TAGN= 010016
L\$CO 002032 G	L10010 024336	OUT51 002260	SKCYL 026302	T\$TEMP= 000000
L\$DEPO 002011 G	L10011 024342	OVRES 017642	SKER 020077	T\$TEST= 000001
L\$DESC 002122 G	L10012 024346	OVWER 020003	SKHS = 000020	T\$TSTM= 177777
L\$DESP 002076 G	L10013 024352	OVWPER 024634	STFLG 003076	T\$TSTS= 000001
L\$DEVP 002060 G	L10014 033674	OVWTRK 003022	STSEC 003230	T\$AUT= 010011
L\$DISP 022464 G	L10015 033752	O\$APTS= 000000	STSEC1 003226	T\$CLE= 010012
L\$DLY 002116 G	MANY 017532	O\$AU = 000000	SURF 003120	T\$DU = 010013
L\$DTP 002040 G	MDHEDR 002000 G	O\$BGNR= 000000	SURFAC 003156	T\$HAR= 010015
L\$DTYP 002034 G	MERGE 031030	O\$BGNS= 000000	SVCGBL= 000000	T\$HW = 010007
L\$DU 024350 G	MID10 002306	O\$DU = 000001	SVCINS= 000000	T\$INI= 010010
L\$DUT 002072 G	MID11 002320	O\$ERRT= 000000	SVCSUB= 177777	T\$MSG= 010005
L\$DVTY 002222 G	MID20 002310	O\$GNSW= 000000	SVCTAG= 000000	T\$PRD= 010006
L\$EF 002052 G	MID21 002322	O\$POIN= 000001	SVCTST= 177777	T\$TES= 010014
L\$ENVI 002044 G	MID30 002312	O\$SETU= 000000	S\$LSYM= 010000	T.DRIV 003060
L\$ETP 002102 G	MID31 002324	PAT = 000006	TEM 003056	T1 032734 G
L\$EXP1 002046 G	MID40 002314	PATLST 003046	TEMP 003066	UAM = 000200 G
L\$EXP4 002064 G	MID41 002326	PNT = 001000 G	TIME 024354	UNLOAD 032222
L\$EXP5 002066 G	MID50 002316	PRI = 002000 G	TQU10 002332	UUT 003130
L\$HARD 033700 G	MID51 002330	PRIOR = 000004	TQU11 002344	VAJMR 030140
L\$HIME 002120 G	MK = 000001	PRI00 = 000000 G	TQU20 002334	VEC = 000002
L\$HPCP 002016 G	MP = 000006	PRI01 = 000040 G	TQU21 002346	VECMG 034010
L\$HPTP 002022 G	NCHECK 030600	PRI02 = 000100 G	TQU30 002336	VECT = 000002
L\$HW 022450 G	NONE 017571	PRI03 = 000140 G	TQU31 002350	VEROD 027612
L\$ICP 002104 G	NXM = 020000	PRI04 = 000200 G	TQU40 002340	VEROW 027226
L\$INIT 022466 G	OBUFF 017236	PRI05 = 000240 G	TQU41 002352	WCOUNT 003142
L\$LADP 002026 G	OFFSET 003146	PRI06 = 000300 G	TQU50 002342	WRITE = 000012
L\$LAST 034042 G	OPI = 002000	PRI07 = 000340 G	TQU51 002354	WRIT1 020136
L\$LOAD 002100 G	OPR001 017302	PRSTRK 003154	TRKCN 003162	WRSEC 025116
L\$LUN 002074 G	OPR002 017321	RDHDR = 000010	TRKFND 003160	XDELAY 017232
L\$MREV 002050 G	OQU10 002262	READ = 000014	TYPDR = 000006	XTIME 024500
L\$NAME 002000 G	OQU11 002274	READ1 020163	T\$ARGC= 000001	X\$ALWA= 000000
L\$PRIO 002042 G	OQU20 002264	REASON 003072	T\$CODE= 002032	X\$FALS= 000040
L\$PROT 022440 G	OQU21 002276	RECER 020023	T\$ERRN= 000322	X\$OFFS= 000400
L\$PRT 002112 G	OQU30 002266	RECMS 017675	T\$EXCP= 000000	X\$TRUE= 000020
L\$REPP 002062 G	OQU31 002300	REGDMP 026550	T\$GMAN= 000000	YDELAY 017234
L\$REV 002010 G				

. ABS. 034042 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28621 WORDS (112 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:16:33

J6

SYMBOL TABLE

CNRLLA.BIN,CNRLLA.LST/-SP-SVC34.MLB/ML,CNRLLA.MAC