

RK611

RK6 FUNCTIONAL CONTROL
CZR6KE0

AH-9130E-MC

MAR 1978

COPYRIGHT © 76-78

digital

FICHE 1 OF 2

MADE IN USA

RK611

RK6 FUNCTIONAL CONTROL
CZR6KE0

AH-9130E-MC

COPYRIGHT © 76-78

FICHE 2 OF 2

MAR 1978

digital

MADE IN USA

This microfiche card contains a grid of frames. The frames on the left side of the card contain data, while the right side is mostly blank. The data in the frames is organized into columns and rows, with some frames containing headers and footers. The data appears to be a list of items or a table of values, but the text is too small to read clearly. The frames are arranged in a regular grid pattern, with approximately 10 columns and 15 rows of frames visible on the left side.

.REM %

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

IDENTIFICATION

PRODUCT CODE: AC-9128E-MC
PRODUCT NAME: CZR6KEO RK6 FCTNL CTRL DIAG
DATE: FEB 1978
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: MARV TEGROTEHUIS

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 DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976, 1978 BY DIGITAL EQUIPMENT CORPORATION

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

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE REQUIREMENTS
2.2	PRELIMINARY PROGRAMS
3.0	OPERATING PROCEDURE
3.1	LOADING PROCEDURE
3.2	STARTUP PROCEDURE
3.3	CONSOLE SWITCH REGISTER
3.4	SOFTWARE SWITCH REGISTERS
3.5	CONTROL C (↑C) OPERATION
3.6	CONTROL S (↑S) OPERATION
3.7	CONTROL Q (↑Q) OPERATION
3.8	UNIBUS ADDRESS
3.9	EXECUTION TIME
4.0	PROGRAM DESCRIPTION
5.0	ERROR REPORTING
6.0	SUBROUTINES

92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147

1.0 ABSTRACT

THE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC (CZR6K) IS A SERIES OF TESTS THAT COMPLETES THE TESTING OF AN RK611/RK06-RK07 SUBSYSTEM. THE DISKLESS CONTROLLER DIAGNOSTIC AND THE RK06-RK07 DRIVE DIAGNOSTIC ARE PREREQUISITES TO THE RUNNING OF THIS PROGRAM. THE PURPOSE OF THIS PROGRAM IS TO TEST THOSE AREAS IN THE CONTROLLER THAT COULD NOT BE TESTED IN A DISKLESS ENVIRONMENT AND THOSE AREAS OF THE DRIVE THAT COULD NOT BE TESTED UNTIL CONTROLLER OPERATION IN A DIAGNOSTIC OR MAINTENANCE MODE HAS BEEN TESTED.

THE TESTS PERFORMED ARE MAINLY FUNCTIONALLY ORIENTED BUT DIAGNOSTIC MODE IS USED IN NUMEROUS OCCASSIONS TO ACCOMPLISH THE OBJECTIVES, MAINLY THE FORCING OF ERRORS. IN THESE CASES, THE CONTROLLER IS PLACED IN DIAGNOSTIC MODE AND OPERATION IS CLOKED PART WAY THROUGH. DIAGNOSTIC MODE IS THEN RESET AND THE CONTROLLER ALLOWED TO COMPLETE THE OPERATION. DEPENDING ON THE OPERATION AND HOW FAR THROUGH IT BEFORE DIAGNOSTIC MODE IS RESET VARIOUS ERROR CONDITIONS CAN BE MADE TO OCCUR. THIS DOCUMENT DOES NOT ATTEMPT TO EXPLAIN WHY THESE ERROR CONDITIONS ARE SET BUT THE INDIVIDUAL TEST DESCRIPTIONS SPECIFY WHAT ERROR IS BEING FORCED AND THE PROCEDURE USED TO FORCE IT.

C A U T I O N

THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING A ↑C. IF THE PROGRAM IS HALTED USING THE HALT KEY THE POSSIBILITY EXISTS THAT THE CARTRIDGE FORMAT WILL BE INCORRECT, THE CYLINDER ADDRESS IN THE DRIVE MAY BE INVALID, OR THE DRIVE MAY NOT BE READY.

2.0 REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

PDP-11 SYSTEM (16K MEMORY)
CONSOLE TERMINAL
DECTAPE, PAPERTAPE, OR DISK
LINE CLOCK (KW11-L) (OPTIONAL)
PARITY OPTION (MM11) (OPTIONAL)
RK611 CONTROLLER
AT LEAST 1 AND UP TO 8 RK06-RK07 DRIVES
FORMATTED RK06K ON EACH DRIVE

2.2 PRELIMINARY PROGRAMS

148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC (ALL PARTS) AND THE

UNIBUS RK06-RK07 DRIVE DIAGNOSTIC (ALL PARTS) SHOULD HAVE RUN SUCCESSFULLY.

3.0 OPERATING PROCEDURE

3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM BINARY TAPE USING THE ABSOLUTE LOADER OR FROM XXDP MEDIA SUPPORTED BY XXDP.

IT CAN BE LOADED AND RUN UNDER APT OR ACT AND IT CAN BE CHAINED BY XXDP.

3.2 STARTUP PROCEDURE

THE PROGRAM START LOCATION IS 200(8). THIS START WILL AUTOMATICALLY SIZE THE SYSTEM UNLESS RUNNING UNDER APT. THE PROGRAM ASSUMES THE STANDARD UNIBUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY LEVEL (177440, 210, AND 4 RESPECTIVELY). IF STARTED AT 200 AND THE XXDP MEDIA IS RK06 (PROGRAM LOADED FROM RK06) DRIVE 0 IS NOT TESTED.

LOCATION 204(8) IS THE PROGRAM RESTART.

LOCATION 214(8) IS THE PARAMETERIZATION START LOCATION. THE OPERATOR WILL BE ASKED TO IDENTIFY THE BUS ADDRESS, VECTOR ADDRESS, AND BUS PRIORITY. IF THE PROGRAM WAS LOADED FROM RK06, THE OPERATOR WILL BE ASKED TO MOUNT A WORK CARTRIDGE ON DRIVE 0 OR TO PLACE IT OFF-LINE IF IT IS NOT TO BE TESTED.

LOCATION 220(8) IS THE PHASE LOCKED LOOP CLOCK ADJUSTMENT START. THE PROGRAM FIRST RUNS THE FIRST THREE TESTS AND THEN JUMPS TO THE ADJUSTMENT ROUTINE. THE PROGRAM WILL CONTINUE TO LOOP IN THIS ROUTINE UNTIL THE PROCESSOR IS HALTED.

ALL DRIVES THAT ARE TO BE TESTED MUST BE ON-LINE, READY, AND WRITE LOCK RESET. IF ALL THREE CONDITIONS ARE NOT MET, THAT DRIVE IS NOT TESTED.

3.3 CONSOLE SWITCH REGISTER

THE CONSOLE SWITCH REGISTER IS USED TO PROVIDE PROGRAM CONTROL AS DESCRIBED BELOW:

SW15 - HALT ON ERROR
SW14 - LOOP ON TEST
SW13 - INHIBIT ERROR REPORT
SW12 - ABORT PROGRAM AFTER 20 ERRORS
SW11 - INHIBIT ITERATIONS
SW10 - BELL ON ERROR
SW09 - LOOP ON ERROR
SW08 - EXECUTE TEST NUMBER SPECIFIED IN SW<7-0>.

SW<7-0> - EXECUTE THIS TEST IF SW08 SET.

EXECUTING A SPECIFIC TEST MUST BE USED WITH CAUTION. SOME TESTS REQUIRE OTHERS TO BE RUN TO FORMAT THE PACK IN A SPECIFIC MANNER OR WRITE SPECIFIC DATA. TESTS THAT REQUIRE OTHERS TO BE RUN INDICATE THIS IN THE TEST DESCRIPTION. IT IS SUGGESTED THAT THE PROGRAM BE RUN IN THE DEFAULT SEQUENCE THE FIRST TIME AFTER IT HAS BEEN LOADED.

NOTE: TEST 3 MUST BE RUN BEFORE ANY SUBSEQUENT TEST. THIS TEST DETERMINES WHICH DRIVES ARE ON THE DRIVE BUS FOR ALL FOLLOWING TESTS. LIKEWISE, TEST 20 MUST BE RUN BEFORE ANY TEST SUBSEQUENT TO IT. THIS TEST READS THE BAD SECTOR FILES AND BUILDS TABLES USED BY THE FOLLOWING TESTS. THESE TESTS, HAVING BEEN RUN ONCE, NEED NOT BE RUN AGAIN IF A DIFFERENT TEST IS SELECTED.

3.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW ='

EACH TIME SWITCH SETTINGS ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROES ARE NOT REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

3.5 CONTROL C (↑C) OPERATION

IF ↑C IS TYPED AT ANY TIME DURING THE PROGRAM EXECUTION THE PROGRAM IS HALTED IMMEDIATELY. IF A MONITOR IS PRESENT (XXDP CHAIN, ACT, APT) THE PROGRAM RETURNS CONTROL TO THE MONITOR. IF NO MONITOR IS PRESENT, THE CPU IS HALTED. DEPRESSING THE CONTINUE KEY WILL DO A PROGRAM RESTART.

3.6 CONTROL S (↑S) OPERATION

IF ↑S IS TYPED AT ANY TIME THE PROGRAM WILL GO INTO A STALL

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259

260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315

LOOP UNTIL A CONTROL Q (↑Q) IS TYPED.

3.7 CONTROL Q (↑Q) OPERATION

IF A ↑S HAS BEEN TYPED, TYPING THE ↑Q CANCELS THE STALL INITIATED BY THE ↑S.

3.8 UNIBUS ADDRESSES

STANDARD UNIBUS ADDRESSES ARE ASSUMED FOR THE KW11-L AND MM11 OPTIONS. THESE ADDRESSES MAY BE CHANGED BY CHANGE THE APPROPRIATE MEMORY LOCATIONS. THE FOLLOWING TAGS AND LOCATIONS HAVE BEEN USED:

KW11-L	TAG	LOCATION
UNIBUS ADDRESS	KWLADD	1710
VECTOR ADDRESS	KWLVEC	1712

3.9 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM FOR ONE DRIVE IS APPROXIMATELY 65 SECONDS AND EACH SUBSEQUENT PASS IS APPROXIMATELY 2 MINUTES 20 SECONDS.

THE EXECUTION TIME FOR MULTIPLE DRIVES IN THE FIRST PASS IS:

((NUMBER OF DRIVES) X 45 SEC) + 20 SEC

FOR SUBSEQUENT PASSES THE RUN TIME IS THE PRODUCT OF 2 MINUTES 20 SECONDS TIMES THE NUMBER OF DRIVES PLUS 25 SECONDS FOR EACH DRIVE AFTER THE FIRST.

4.0 PROGRAM DESCRIPTION

THE FOLLOWING TEST SEQUENCE IS EXECUTED ASSUMING TWO OR MORE DRIVES.

FIRST PASS - FIRST DRIVE:
ALL TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE.

FIRST PASS - ALL REMAINING DRIVES:
STATUS VALID TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH DRIVE.

THEN MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH COMBINATION OF DRIVES.

SECOND AND ALL SUBSEQUENT PASSES:
THE SAME SEQUENCE OF TESTING IS REPEATED EXCEPT FOR TEST ITERATIONS WHICH ARE SPECIFIED FOR EACH TEST.

316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371

**BASIC INTERFACE AND OPTION TESTS

TEST 1 RK611 BASE ADDRESS TEST

CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT CAUSE A NON-EXISTANT MEMORY TRAP.

TEST 2 INTERRUPT VECTOR ADDRESS TEST CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE EXPECTED ADDRESS.

**STATUS VALID TESTS

TEST 3 SELECT ALL DRIVES

IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT, DETERMINE WHAT DRIVES ARE ON-LINE BY SELECTING ALL DRIVES. IF NON-EXISTENT DRIVE REPORTED MAKE SURE STATUS VALID IS RESET. IF DRIVE PRESENT MAKE SURE NO ERROR EXISTS, DRIVE IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.

IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE CYCLED UP, AND STATUS VALID SET.

IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM START (214) WAS USED. IF THE AUTOMATIC START (200) IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204) WILL RETAIN THE TEST STATUS OF DRIVE 0.

IF THE PARAM START IS USED, THE OPERATOR MUST EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.

ALL DRIVES TO BE TESTED MUST BE ON-LINE, CYCLED UP, AND WRITE LOCK RESET. ADDRESSES OF DRIVES THAT ARE NON-EXISTANT EITHER BECAUSE THE DRIVE DOES NOT EXIST OR IS OFF-LINE ARE USED TO VERIFY NON-EXISTANT DRIVE ERROR DETECTION. DRIVES THAT ARE ON-LINE BUT NOT CYCLED UP OR ARE WRITE LOCKED ARE NOT TESTED.

AT COMPLETION OF THE TEST A MESSAGE WILL BE GIVEN TO IDENTIFY THE DRIVES TO BE USED IN TESTING.

NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE ANY OTHER TEST THAT FOLLOWS.
THE DETERMINATION OF WHETHER EACH DRIVE IS AN RK06 OR RK07 IS MADE IN THIS TEST, AND

A TABLE IS FILLED ACCORDINGLY.

TEST 4 RELEASE ALL DRIVES

RELEASE ALL DRIVES. MAKE SURE NO ERROR SETS AND STATUS VALID IS RESET.

TEST 5 NON-STANDARD MESSAGES AND SVAL

PICK ONE OF THE AVAILABLE DRIVES AND GET NON-STANDARD MESSAGES. MAKE SURE NO ERROR OCCURS AND STATUS VALID DOES NOT SET AND THAT NON-STANDARD MESSAGES CAUSE STATUS VALID TO RESET.

TEST 6 WRITING CS2 AND STATUS VALID

SELECT AN AVAILABLE DRIVE. MAKE SURE STATUS VALID IS SET. WRITE COMMAND AND STATUS REGISTER 2. MAKE SURE STATUS VALID RESETS.

**CONTROLLER ERROR TESTS

TEST 7 DRIVE TYPE ERROR

CREATE A DRIVE TYPE ERROR MAKE SURE DRIVE TYPE ERROR SETS AND STATUS VALID SETS.

TEST 10 STATUS VALID AND PARITY ERROR

ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY. MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION, DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT, AND STATUS VALID SET. ISSUE A CONTROLLER CLEAR. MAKE SURE DRIVE INTERRUPT AND ATTENTION ARE STILL SET. SELECT DRIVE AGAIN WITH GOOD PARITY. MAKE SURE ATTENTION, DRIVE STATUS CHANGE, DRPAR, CONTROLLER ERROR, DRIVE INTERRUPT, AND STATUS VALID ARE SET AND SPAR IS RESET. ISSUE A CONTROLLER CLEAR. GET NON-STANDARD MESSAGES AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION ARE SET. CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT FOR ALL AVAILIABLE DRIVES.

TEST 11 UNIT FIELD ERROR ON RELEASE

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A RELEASE COMMAND. CLOCK THROUGH PHASE ADDRESS 2. TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD ERROR SETS.

TEST 12 UNIT FIELD ERROR ON SELECT

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE.

372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427

PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A SELECT
COMMAND WITH MESSAGE ID = 3 AND DRIVE SELECTED = 0.
CLOCK THROUGH PHASE ADDRESS 6. TURN OFF DIAGNOSTIC

MODE. MAKE SURE UNIT FIELD ERROR SETS.

**ATTENTION HANDLING BY CONTROLLER

TEST 13 DOUBLE INTERRUPT

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. MAKE
SURE STATUS VALID IS SET. CHECK THAT SECOND INTERRUPT
OCCURS. AFTER SECOND INTERRUPT CHECK THAT STATUS
VALID IS RESET. ISSUE SELECT AND MAKE SURE STATUS
VALID SETS. CLEAR DRIVE. CHECK THAT DRIVE STATUS
CHANGE SETS (BIT 14 OF DRIVE STATUS REGISTER)

TEST 14 SINGLE INTERRUPT FROM ATTENTION

DO A SEEK TO CYLINDER 0. WAIT FOR INTERRUPT FROM
DRIVE ATTENTION. LOWER PRIORITY AGAIN AND MAKE SURE
ANOTHER INTERRUPT DOES NOT OCCUR. CLEAR DRIVE.

TEST 15 RESET ATTENTIONS WITH UNIBUS INIT

DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.

**ILLEGAL DISK ADDRESS ERROR TESTS

TEST 16 ILLEGAL DISK ADDRESS (PART 1)

ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE ILLEGAL
ADDRESS ERROR AND SEEK INCOMPLETE SETS. CLEAR
CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7.
CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.

TEST 17 ILLEGAL DISK ADDRESS (PART 2)

ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
ILLEGAL DISK ADDRESS ERROR SETS. CLEAR CONTROLLER AND
DRIVE
THIS TEST IS BYPASSED BY THE RK07 CONTROLLER BECAUSE
IT DOES NOT RECOGNIZE ANY ILLEGAL DISK ADDRESS.

**WRITE HEADER TESTS

428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483

TEST 20 READ BAD SECTOR INFORMATION

ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632, TRACK 2 TO GET THE FACTORY DETECTED BAD SECTOR FILE, 26 SECTOR MODE. CYLINDER 1456 IS USED FOR THE RK07.

IF AN ERROR OCCURS, READ SECTOR 2, 4, 6, OR 10(8) UNTIL A SUCCESSFUL READ IS DONE. IF NONE READ SUCCESSFULLY REMOVE THIS DRIVE FROM TEST. WHEN A READ IS SUCCESSFUL, TEST THAT THE PACK IS NOT AN ALIGNMENT PADK AND STORE THE ENTRIES FOR LATER USE.

REPEAT THIS SERIES OF OPERATIONS FOR FACTORY DETECTED BAD SECTORS 24 SECTOR MODE, SOFTWARE DETECTED BAD SECTORS 26 SECTOR MODE, AND SOFTWARE DETECTED BAD SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR 24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED FROM TESTING.

NOTE: THIS TEST IS RUN IN THE FIRST (QUICK VERIFY) PASS ONLY.

TEST 21 FORMAT IN 26 SECTOR FORMAT

FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER AFTER READ HEADER.

**HEADER RECOGNITION TESTS

TEST 22 BAD SECTOR ERROR

FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1 (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS AND ALL OTHER SECTORS GOOD.

ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS. ISSUE A WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS. MAKE SURE BAD SECTOR ERROR SET. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2. MAKE SURE NO ERROR SETS.

TEST 23 HEADER VRC ERROR

FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE 16 SECTORS WITH BAD HEADER VRC. ISSUE A WRITE DATA OF EACH OF THE SECTORS WITH A BAD HEADER VRC. MAKE SURE HEADER VRC ERROR SETS. ISSUE A WRITE DATA TO A GOOD

484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595

HEADER AND MAKE SURE NO ERROR OCCURS.

TEST 24 BAD SECTOR ERROR AND HVRC ERROR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS BOTH A BAD SECTOR ERROR AND HEADER VRC. ISSUE A WRITE

DATA TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE ONLY HEADER VRC ERROR SETS.

TEST 25 OPERATION INCOMPLETE

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE WRONG FORMAT. ISSUE A WRITE DATA OF 400 TO CYLINDER 0, TRACK 0, SECTOR 21. MAKE SURE OPI SET.

TEST 26 OPI WITH HVRC ERROR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0, SECTOR 17. THAT BOTH OPERATION INCOMPLETE AND HEADER VRC SET.

TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR

FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0, AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE END OF THE OPERATION.

**DATA TRANSFER TESTS

TEST 30 WRITE AND READ ONE SECTOR

FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 2 TO AGREE WITH BAD SECTOR INFORMATION. ISSUE A WRITE DATA OF ONE SECTOR ON CYLINDER 312, TRACK 0. READ IT BACK TO MAKE SURE IT AGREES WITH WHAT IS WRITTEN.

TEST 31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT

ISSUE A WRITE DATA OF ONE SECTOR TO CYLINDER 312, TRACK 2, SECTOR 12 WITH INHIBIT BUS ADDRESS INCREMENT. READ DATA BACK TO MAKE SURE EVERY WORD IS THE SAME AND CORRECT.

TEST 32 WRITE DATA ADDRESS GREATER THAN 32K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K OF
MEMORY IS PRESENT.

TEST 33 READ DATA ADDRESS GREATER THAN 32K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 177770.
CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K OF
MEMORY IS PRESENT.

TEST 34 WRITE DATA ADDRESS GREATER THAN 64K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K OF
MEMORY IS PRESENT.

TEST 35 READ DATA ADDRESS GREATER THAN 64K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K OF
MEMORY IS PRESENT.

TEST 36 WRITE DATA ADDRESS GREATER THAN 96K

ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
MAKE SURE CORRECT DATA IS ON DISK.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K OF
MEMORY IS PRESENT.

TEST 37 READ DATA ADDRESS GREATER THAN 96K

ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
CHECK MEMORY FOR CORRECT TRANSFER.

NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K OF
MEMORY IS PRESENT.

TEST 40 PARTIAL SECTOR WRITE DATA

ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312, HEAD
0, SECTOR 0. MAKE SURE THE SECTOR WAS ZERO FILLED
CORRECTLY.

TEST 41 PARTIAL SECTOR READ DATA

596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651

652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707

WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A KNOWN CONFIGURATION. ISSUE A READ DATA OF 103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE ONLY 103 WORDS GET TRANSFERRED TO MEMORY.

TEST 42 WRITE DATA WITH NON-EXISTENT MEMORY

ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000. MAKE SURE NON-EXISTENT MEMORY SETS.

TEST 43 READ DATA WITH NON-EXISTENT MEMORY

ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000. MAKE SURE NON-EXISTENT MEMORY SETS.

TEST 44 UNIBUS PARITY ERROR

INITIALIZE A MEMORY LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF 400 WORDS STARTING AT A LOCATION 112 WORDS BEFORE THE LOCATION WITH BAD PARITY. MAKE SURE THAT UNIBUS PARITY ERROR SETS.

NOTE: THIS TEST IS ONLY EXECUTED IF MEMORY PARITY OPTION EXISTS FOR BUFFER.

**MULTIPLE SECTOR OPERATIONS

TEST 45 TWO SECTOR WRITE DATA (PART 1)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE IT IS CORRECT.

TEST 46 TWO SECTOR WRITE DATA (PART 2)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 23. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID NOT TAKE PLACE.

TEST 47 TWO SECTOR WRITE DATA (PART 3)

ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312, TRACK 0, SECTOR 10. READ DATA BACK ONE SECTOR AT A TIME AND CHECK ZERO FILL OF SECOND SECTOR.

TEST 50 MID-TRANSFER SEEK ON WRITE (PART 1)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 25. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763

TEST 51 MID-TRANSFER SEEK ON WRITE (PART 2)

ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 2, SECTOR 25. READ DATA BACK ONE SECTOR AT A TIME AND MAKE SURE A MID-TRANSFER SEEK DID TAKE PLACE.

TEST 52 TWO SECTOR READ DATA (PART 1)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS READ.

NOTE: TWO SECTOR WRITE DATA (PART 1) MUST BE EXECUTED BEFORE THIS TEST.

TEST 53 TWO SECTOR READ DATA (PART 2)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 23. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.

NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE EXECUTED BEFORE THIS TEST.

TEST 54 TWO SECTOR READ DATA (PART 3)

ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312, TRACK 0, SECTOR 10. VERIFY THAT ALL 401 WORDS ARE PLACED IN MEMORY.

NOTE: TWO SECTOR WRITE DATA (PART 3) MUST BE EXECUTED

BEFORE THIS TEST.

TEST 55 MID-TRANSFER SEEK ON READ (PART 1)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 25. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES OCCUR.

NOTE: MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE EXECUTED BEFORE THIS TEST.

TEST 56 MID-TRANSFER SEEK ON READ (PART 2)

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 2, SECTOR 25. VERIFY THAT CORRECT DATA IS READ AND A MID-TRANSFER SEEK DOES OCCUR.

NOTE: MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE EXECUTED BEFORE THIS TEST.

TEST 57 CYLINDER ADDRESS OVERFLOW (PART 1)

764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819

ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632, TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS OVERFLOW ERROR DOES NOT OCCUR. CYLINDER 1456 IS USED FOR THE RK07.

TEST 60 CYLINDER ADDRESS OVERFLOW (PART 2)

ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632, TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS OVERFLOW ERROR DOES OCCUR. CYLINDER 1456 IS USED FOR THE RK07.

**18 BIT DATA TRANSFER TESTS

TEST 61 FORMAT IN 24 SECTOR FORMAT

FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER AFTER READ HEADER.

TEST 62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0, SECTOR 0. READ SECTOR BACK AND MAKE SURE IT IS CORRECT.

TEST 63 24 SECTOR FORMAT DATA TRANSFER (PART 2)

LOAD A LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312,

TRACK 0, SECTOR 0 WITH BUFFER BEGINNING 112 WORDS BEFORE WORD WITH BAD PARITY. MAKE SURE UNIBUS PARITY ERROR DOES NOT SET. READ SECTOR BACK AND MAKE SURE IT IS CORRECT.

NOTE: THIS TEST IS EXECUTED ONLY IF MEMORY PARITY EXISTS FOR SPECIFIED LOCATION.

TEST 64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0, SECTOR 23. READ SECTOR BACK AND MAKE SURE IT IS CORRECT. MAKE SURE THAT MID-TRANSFER SEEK HAS TAKEN PLACE.

**SPECIAL DATA TRANSFER TESTS

TEST 65 MULTI-SECTOR DATA TRANSFER AND BSE

820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875

FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH SECTOR 1 MARKED BAD. ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS AND RKDA IS CORRECT. READ SECTOR 0 AND MAKE SURE IT IS CORRECT.

ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS AND THE PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.

TEST 66 FORMAT TEST

FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT. MAKE SURE NO ERRORS SET. READ SECTORS 0-25 AND MAKE SURE DATA CHECK DOES NOT OCCUR.

**WRITE CHECK TESTS

TEST 67 WRITE-CHECK WITH NO ERROR

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN. DO A WRITE-CHECK OF 400 WORDS. MAKE SURE NO ERROR OCCURS.

TEST 70 WRITE CHECK ERROR (PART 1)

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING CONFIGURATIONS:

000001 000020 000400 010000
000002 000040 001000 020000
000004 000100 002000 040000
000010 000200 004000 100000

MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT IS CORRECT.

TEST 71 WRITE CHECK ERROR (PART 2)

WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777 IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS ONE OF THE FOLLOWING CONFIGURATIONS:

177776 177757 177377 167777
177775 177737 176777 157777
177773 177677 175777 137777
177767 177577 173777 077777

MAKE SURE WRITE CHECK ERROR SET FOR EACH OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS AND WORD COUNT IS CORRECT.

TEST 72 WRITE CHECK OF PARTIAL SECTOR

WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF 110 WORDS MAKING SURE THE 111TH WORD IS DIFFERENT THAN DATA ON DISK. MAKE SURE WRITE CHECK ERROR DOES NOT SET.

**MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT

TEST 73 MAXIMUM DATA TRANSFER (PART 1)

IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO INSURE THE FORMAT IS CORRECT.

ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH ONE SECTOR WRITES. ISSUE A SEEK TO CYLINDER 0, TRACK 0. ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 20000 WORDS TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE CONTROLLER TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS, DISK ADDRESS, BUS ADDRESS AND WORD COUNT. READ EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.

NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT IN THE FIRST 256 SECTORS ON THE PACK.

TEST 74 MAXIMUM DATA TRANSFER (PART 2)

ZERO OUT FIRST 256 SECTORS OF THE DISK WITH 20000 WORD WRITE. SEEK TO CYLINDER 632. ISSUE A WRITE OF MAXIMUM DATA TRANSFER 20000 WORD WRITE. MAKE SURE CONTROLLER TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS DISK ADDRESS, BUS ADDRESS AND WORD COUNT. SEEK TO CYLINDER 632. ISSUE A WRITE CHECK OF 20000 WORDS. MAKE SURE NO ERROR SETS.

NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT IN THE FIRST 256 SECTORS ON THE PACK.
CYLINDER 1456 IS USED FOR THE RK07.

TEST 75 CONTROLLER TIME OUT

876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931

SEEK TO CYLINDER 632. ISSUE A RECALIBRATE AND DO NOT
WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER OF
CYLINDER 0, TRACK 0. MAKE SURE CONTROLLER TIME OUT
SETS.
CYLINDER 1456 IS USED FOR THE RK07.

**ERRORS DURING DATA TRANSFER

TEST 76 LIMIT DETECT ON DATA TRANSFER

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
A SEEK TO CYLINDER 2 WITH BAD PARITY. ISSUE A DRIVE
CLEAR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
TRACK 0, HEAD 0. SEEK INCOMPLETE BECAUSE OF OUTER

TEST 77 PROGRAMMING ERROR

ISSUE A SUBSYSTEM CLEAR. ISSUE A READ DATA OF 400
WORDS ON CYLINDER 0, TRACK 0, SECTOR 0. DURING READ
ISSUE A WRITE TO THE SPARE REGISTER. MAKE SURE
PROGRAMMING ERROR SETS.

TEST 100 ECC HARD

ISSUE A SUBSYSTEM CLEAR. ISSUE A WRITE DATA WORDS
CONSISTING OF 177777 TO CYLINDER 0, TRACK 0, SECTOR 0.
NOW WRITE ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0.
DURING WRITE ISSUE CONTROLLER CLEAR. MAKE SURE
PROGRAMMING ERROR IS RESET. NOW ISSUE A READ DATA TO
CYLINDER 0, TRACK 0, HEAD 0 AND AN ECC HARD ERROR
SHOULD SET.

TEST 101 DRIVE TIMING ERROR

ISSUE A SUBSYSTEM CLEAR. SEEK TO CYLINDER 632. ISSUE
A RECALIBRATE BUT DO NOT WAIT FOR SECOND INTERRUPT.
PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ
HEADER OF CYLINDER 0, TRACK 0. CLOCK THROUGH SEEK AND
DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE.
DRIVE TIMING ERROR SHOULD SET BECAUSE OF NO DATA
TRANSISTIONS ON DATA LINE.

**ERROR FORCING IN DRIVE

TEST 102 INITIALIZE CLEARING SACK

ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE DRIVE.
ISSUE A SUBSYSTEM CLEAR. PUT CONTROLLER IN DIAGNOSTIC
MODE. ISSUE A SELECT COMMAND WITH MESSAGE ID = 3 AND
DRIVE SELECTED = 0. CLOCK THROUGH PHASE ADDRESS 6.

932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
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

TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD ERROR DOES NOT SET.

TEST 103 DRIVE OFF TRACK

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0, TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILABLE DRIVES.

TEST 104 FILE UNSAFE

ISSUE A SUBSYSTEM CLEAR. ISSUE A RECLAIBRATE. ISSUE A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR FORMAT. DO A SELECT COMMAND IN 26 SECTOR FORMAT. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR FORMAT. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES. SIMULATE INDEX PULSE. TURN OFF DIAGNOSTIC MODE. FILE UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE THROUGH SECTOR PULSE. REPEAT FOR ALL AVAILABLE DRIVES.

TEST 105 DUMMY TEST FOR PREVIOUS TEST EXIT

THIS TEST IS PRESENT TO MAKE \$SWO8TB TABLE HAVE AN ENTRY WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST. IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".

IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE

ALLOWED TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS FROM THE DRIVE BECOMING READY AT A LATER TIME.

**MULTI-DRIVE OPERATIONS

TEST 106 RESET ATTENTIONS WITH UNIBUS INIT

DO A RECALIBRATE ON ALL AVAILIABLE DRIVES. ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.

TEST 107 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

DO A RECALIBRATE ON ALL AVAILIABLE DRIVES. ISSUE A SUBSYSTEM CLEAR. MAKE SURE ALL ATTENTIONS RESET.

988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
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

TEST 110 SVAL AND ATTENTION FROM OTHER DRIVE

DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS CHANGE REMAINS RESET.

REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.

NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST TWO DRIVES ARE AVAILABLE.

TEST 111 OVERLAPPED OPERATIONS

DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B. AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD BE SET AND DRIVE A HAS ATTENTION SET.

REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.

NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE TEST WILL NOT BE DONE.

5.0 ERROR REPORTING

A DETAILED DESCRIPTION OF THE ERROR FORMATS AND REPORTS CONTENTS IS GIVEN HERE. THIS IS ESSENTIALLY THE SAME AS CAN BE FOUND IN THE LISTING COMMENTS UNDER THE ERROR POINTER TABLE.

ERROR POINTER TABLE:

THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR. THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.

NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).

NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS

EXPLAINED AS FOLLOWS:

EM ::POINTS TO THE ERROR MESSAGE

DH ::POINTS TO THE DATA HEADER

DT ::POINTS TO THE DATA

DF ::POINTS TO THE DATA FORMAT

EM AND DH ARE ASCIZ DATA. EM IS ALWAYS A MESSAGE BUT DH CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACCROSS THE PAGE. DT IS A STRING OF WORDS THAT POINT TO THE DATA TO BE TYPED, AND DF IS A STRING OF WORDS THAT TELL HOW THE DT WORDS ARE TO BE TYPED. IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A PARTICULAR FORMAT, IT IS REPLACED WITH A ZERO. THE NORMAL USAGE OF THE ERROR TABLE IS TO HAVE A TABLE ENTRY FOR EACH ERROR MESSAGE THAT CAN OCCUR. IN THE INTEREST OF ECONOMICS OF

1044
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
1096
1097
1098
1099

CORE MEMORY. THIS PROGRAM USES THE ERROR TABLE IN A SLIGHTLY DIFFERENT MANNER AS DESCRIBED BELOW.

THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT FORMATS THAT ARE USED THROUGHOUT THE PROGRAM. WHEN AN ERROR IS TO BE REPORTED, THE TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT IS CHOSEN FROM THE DEFINED SET. THE TABLE ENTRY CHOSEN IS THEN ALTERED BY CHANGING THE FIRST (AND POSSIBLY THE SECOND) WORD TO CONTAIN THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE MESSAGE PORTION OF THE REPORT. THE DATA FIELDS FOR THAT ENTRY ARE NEVER CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.

THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW WITH THE DEFINITION OF THE ENTRY. ALL DATA FIELDS ARE TYPED IN OCTAL.

ERROR ITEM 1
(MESSAGE)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC DRVNUM

ERROR ITEM 2
(MESSAGE)
(MESSAGE)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
RKBA RKWC
T.BA T.WC

ERROR ITEM 3
(MESSAGE)
TST NUM ERR PC DRIVE

\$TESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKMRI
T.CS1 T.CS2 T.DS T.ER T.ASF T.MRI

ERROR ITEMS 4, 5, 6, AND 7 ARE USED TO REPORT ERRORS THAT ARE THE RESULT OF A HARDWARE ERROR INDICATOR BEING SET WHEN NOT EXPECTED, NOT SET WHEN IT IS EXPECTED, OR BOTH. THE ERROR REPORT WILL CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2) ALL THE EXPECTED ERRORS THAT DID NOT OCCUR, OR (3) ALL THE EXPECTED BUT NOT SET ERRORS AND THE UNEXPECTED BUT SET ERRORS.

THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT. INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING STATEMENTS:

"THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
"THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
"THE ABOVE ARE ERRORS SET IN OPERATION:"

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
1151
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
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211

PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT SPECIFY THE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE THAT IDENTIFIES THE OPERATION BEING PERFORMED.

EXAMPLE:
NON-EXISTANT DRIVE
THE ABOVE ARE ERRORS SET IN OPERATION:
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.

EXAMPLE:
NON-EXISTANT DRIVE
THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
DRIVE SELECT

(ADDITIONAL LINES OF INFORMATION)

THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR, I.E. A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.

EXAMPLE:
NON-EXISTANT MEMORY
THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
UNIBUS PARITY ERROR
THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
WRITE DATA

(ADDITIONAL LINES OF INFORMATION)

THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN UNIBUS PARITY ERROR WAS EXPECTED.

ERROR ITEM 4
(DESCRIPTION OF ERROR)
ERROR IN OPERATION
(DESCRIPTION OF OPERATION)
TST NUM ERR PC DRIVE
STESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
RKBA RKWC
T.BA T.WA
A00 B00 A01 B01 A02 B02 A03 B03
\$REG10 \$REG11 \$REG12 \$REG13 \$REG14 \$REG15 \$REG16 \$REG17

THE ERRORS REPORTED BY THIS FORMAT ARE:
CONTROLLER DETECTED DRIVE BUS ERROR
DRIVE DETECTED DRIVE BUS ERROR
SEEK INCOMPLETE
NON-EXECUTABLE DRIVE FUNCTION
DRIVE TIMING ERROR

DRIVE UNSAFE
AC LOW
SPINDLE SPEED LOSS
DRIVE OFF TRACK
ILLEGAL DRIVE ADDRESS ERROR
CYLINDER OVERFLOW
DRIVE TYPE ERROR
FORMAT ERROR
WRITE LOCK ERROR

ERROR ITEM 5
THIS ENTRY IS THE SAME AS ITEM 4 WITH THE ADDITION OF A
MESSAGE THAT FOLLOWS. THIS MESSAGE IS:

"ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"

THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR ADD
THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.

ERROR ITEM 6

(DESCRIPTION OF ERROR)
ERROR IN OPERATION
(DESCRIPTION OF OPERATION)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
RKBA RKWC
T.BA T.WC

THE ERRORS REPORTED BY THIS FORMAT ARE:
DATA CHECK
WRITE CHECK

ECC HARD
DATA LATE
OPERATION INCOMPLETE
HEADER VRC ERROR
BAD SECTOR ERROR

ERROR ITEM 7

(DESCRIPTION OF ERROR)
ERROR IN OPERATION
(DESCRIPTION OF OPERATION)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF
T.CS1 T.CS2 T.DS T.ER T.ASOF

THE ERRORS REPORTED BY THIS FORMAT ARE:
NON-EXISTANT DRIVE
NON-EXISTANT MEMORY
CONTROLLER TIME OUT

1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267

UNIT FIELD ERROR
MULTIPLE DRIVE SELECT
PROGRAMMING ERROR
UNIBUS PARITY ERROR
ILLEGAL FUNCTION CODE

DESCRIPTION OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL.

ERROR ITEM 10
(DESCRIPTION OF ERROR)
ERROR AT COMPLETION OF OPERATION
(DESCRIPTION OF OPERATION)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC \$DRVNUM
EXPT IS
\$REG10 \$REG11

THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS. THE SPECIFIC ERRORS ARE:

WORD COUNT INCORRECT
BUS ADDRESS INCORRECT
CYLINDER ADDRESS INCORRECT
TRACK ADDRESS INCORRECT
SECTOR ADDRESS INCORRECT

ERROR ITEM 11
(ERROR INDICATOR OR STATUS BIT)
NOT SET AS A RESULT OF
(ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC \$DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKMRI

T.CS1 T.CS2 T.DS T.ER T.ASOF T.MRI

ERROR ITEM 12
THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
"NOT RESET AS A RESULT OF"

ERROR ITEM 13
THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
"SET AS A RESULT OF"

ERROR ITEM 14
THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
"RESET AS A RESULT OF"

ERROR ITEM 15
(HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)

1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323

1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379

TST NUM ERR PC DRIVE
STESTN SERRPC DRVNUM
GOOD BAD WORD NUM
\$REG10 \$REG11 \$REG12

ERROR ITEM 16
ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15.

6.0 SUBROUTINES

IN THE INTEREST OF CONSERVING MEMORY, IT IS NECESSARY TO MAKE EXTENSIVE USE OF SUBROUTINES. HOWEVER, IN THE INTEREST OF PRESERVING CODE READABILITY, SUBROUTINE NAMING IS DESCRIPTIVE OF THE FUNCTION PERFORMED. THE SUBROUTINE FUNCTION IS KEPT SMALL AND IN GENERAL A SUBROUTINE ONLY PERFORMS ONE FUNCTION, I.E., LOAD THE RK611 REGISTER AND START AN OPERATION (TLOADRK) OR WAIT A SPECIFIED NUMBER OF MILLISECONDS FOR AN INTERRUPT (TWATNN WHERE NN VARIES FROM CALL TO CALL AND IS THE TIME TO WAIT). THE FOLLOWING IS A DESCRIPTION OF THE SUBROUTINES NOT PROVIDED BY SYSMAC:

LINE CLOCK CALIBRATE

WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE LINE CLOCK SIMULATOR.

CALL: JSR PC,CLKCAL

OPTION PRESENT TEST AND SETUP

THIS ROUTINE CHECKS IF THE MEMORU PARITY OPTION AND THE LINE CLOCK ARE ON THE SYSTEM. FLAGS ARE SET IF PRESENT; CLEARED OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.

CALL: JSR PC,OPTTST

LINE CLOCK SIMULATION ROUTINE

THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO DO THIS THE VALUE STORED IN MILCNT IS USED AS THE BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT IS

1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435

RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK TICK).

THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.

CALL: JSR PC,MYTIME

WAIT FOR INTERRUPT ROUTINE

THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED (BUT NOT PRINTED IN THE ROUTINE) AND THEN RETURNS TO THE LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.

CALL: TWAT16 THROUGH TWAT159, TWAT1S, TWAT2S, TWAT8S, AND TWAT1M

"L" REGISTER LOADING ROUTINE

THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.

CALL: JSR R4,LRLOAD
COMMAND
WORD COUNT
BUS ADDRESS
.BYTE SECTOR ADDRESS
.BYTE TRACK ADDRESS
CYLINDER ADDRESS

LOAD RK611 FOR OPERATION

THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER. THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION OF THE REGISTER CONTENTS. L.CS1 IS TRANSFERRED LAST AS IT SHOULD BE IF THE GO BIT IS SET.

CALL: TLOADRK

1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491

STORE RK611 REGISTERS

ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT
STORED IN THIS ROUTINE.

CALL: TGETRK

BIT COUNTER IN A WORD

THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE
STACK BY THE CALLING ROUTINE. THE NUMBER OF BITS
FOUND IN THE WORD ARE PASSED BACK ON THE STACK.

CALL: JSR R4,BITCNT

MAINTENANCE CLOCK ROUTINE

THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN
THE ADDRESS AFTER THE CALL. THE FIRST BYTE CONTAINS
THE NUMBER OF PHASE ADDRESSES THE CALLING ROUTINE
WANTS THE CONTROLLER CLOCKED THROUGH AND THE SECOND
BYTE CONTAINS THE NUMBER OF CLOCK TRANSITIONS (PARTIAL
PHASES) TO BE DONE.

CALL: JSR R4,MCLOCK
.BYTE ;NUMBER OF CLOCK TRANSISTIONS

.BYTE ;NUMBER OF PHASE ADDRESSES

READ AND SORT HEADERS

THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY THE
FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
ASCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN
THE READ HEADER OPERATION AND DATA LATE IS CHECKED
AFTER EACH READ OF THE DATA BUFFER.

CALL: JSR R4,RDSTHD
TCHKOP ;RETURN POINT IF CERR IN READ
;HDR
ERROR 4 ;OR 5, 6, 7
ERROR 13 ;RETURN IF DATA LATE IN DB
;UNLOAD
ERROR 2 ;RETURN IF TO SLOW OR

1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547

;IF HDR 0 NOT FOUND

GET DRIVE STATUS

THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT
IN \$REG10 THROUGH \$REG17. THESE REGISTORS ARE FIRST
CLEARED TO ALL ONES AND THEN IF ERROR OCCURS WHILE
GETTING STATUS, THE 1'S ARE LEFT IN THE REGISTERS.

CALL: JSR R4,GETDRS
BR ERROR PROCESSING ERROR RETURN
BR NO ERROR PROCESSING GOOD RETURN

SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST

THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
COMMAND. CERR AND DI ARE MONITORED FOR A SHORT PEIROD
OF TIME DURING WHICH THEY SHOULD BOTH RESET.

IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.

IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN
APPROPRIATE ERROR MESSAGE IS PREPARED AND REPORTED
WHEN THE ROUTINE RETURN TO THE CALL. IF EVERY THING
IS GOOD, THE RETURN SKIPS OVER THE ERROR CALL AND TEST
ABORT.

THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY AN
ERROR MESSAGE AND BRANCH TO END OF TEST. THIS IS DONE
BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO

THE TEST.

CALL: TSSINIT

WORD COUNT AT END OF OPERATION CHECK

THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE
FOR THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF
UNEQUAL, THE ERROR FLAG (WCERR) IS SET IN GROUP 4
ERROR FLAGS (GRP4ER)

CALL: JSR R4,CHKWC
.WORD ;EXPECTED WC VALUE

BUS ADDRESS AT END OF OPERATION CHECK

1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603

THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT. IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)

IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS ADDRESS IS THE STARTING BUS ADDRESS.

CALL: JSR R4,CHKBA

CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION

THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE OPERATION. THE CONTENTS OF THE LOAD REGISTER STORAGE ARE COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE NECESSARY CALCULATION.

ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES STORAGE EXPCYL, EXPTRK, EXPSEC. ALL 3 ARE CHECKED AND IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE ERROR FLAGS FIELD (GRP4ER) IS SET.

CALL: JSR R4,CHKCTS

OPERATION CHECK ROUTINE

THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME SOFTWARE ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS: THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL BE THE PROPER ITEM FOR THE FORMAT REQUIRED BY THIS ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4, 5, 6, 7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT FORMAT.

FOR NO EXPECTED ERRORS:
CALL: TCHKOP

FOR EXPECTED ERRORS:
CALL: TCHKWE
.WORD ;GROUP 1 EXPECTED ERRORS

E03

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 30

SEQ 0030

1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659

.WORD ;GROUP 2 EXPECTED ERRORS
.WORD ;GROUP 3 EXPECTED ERRORS

WHERE EACH BIT IN THE THREE WORDS FOLLOWING THE CALL
REPRESENT A SPECIFIC ERROR. THE BIT ASSIGNMENTS ARE
GIVEN BELOW:

GROUP 1 ERRORS:
BIT 0 - CONTROLLER DETECTED DRIVE BUS
PARITY ERROR
BIT 1 - SEEK INCOMPLETE
BIT 2 - NON-EXECUTABLE DRIVE FUNCTION
BIT 3 - DRIVE DETECTED DRIVE BUS PARITY ERROR
BIT 4 - FORMAT ERROR
BIT 5 - DRIVE TYPE ERROR
BIT 6 - AC LOW ERROR
BIT 7 - SPEED LOSS ERROR
BIT 8 - DRIVE OFF TRACK ERROR
BIT 9 - CYLINDER OVERFLOW ERROR
BIT 10 - ILLEGAL DISK ADDRESS ERROR
BIT 11 - WRITE LOCK ERROR
BIT 12 - DRIVE TIMING ERROR
BIT 13 - NO CERR WITH OTHER ERROR SET ERROR
BIT 14 - DRIVE UNSAFE ERROR
BIT 15 - CERR BUT NO OTHER ERROR SET ERROR

GROUP 2 ERRORS:
BIT 0 - ECC HARD ERROR
BIT 1 - DATA CHECK ERROR
BIT 2 - WRITE CHECK ERROR
BIT 3 - DATA LATE ERROR
BIT 4 - OPERATION INCOMPLETE ERROR
BIT 5 - HEADER VRC ERROR
BIT 6 - BAD SECTOR ERROR

GROUP 3 ERRORS:
BIT 0 - NON-EXISTANT DRIVE ERROR
BIT 1 - CONTROLLER TIMEOUT ERROR
BIT 2 - UNIT FIELD ERROR
BIT 3 - MULTIPLE DRIVE SELECT ERROR
BIT 4 - PROGRAMMING ERROR
BIT 5 - NON-EXISTANT MEMORY ERROR
BIT 6 - UNIBUS PARITY ERROR
BIT 7 - ILLEGAL FUNTION ERROR

BAD SECTOR CHECK

THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED
THEREIN THAT HAVE THE CYLINDER AND TRACK ADDRESS
SPECIFIED IN L.DCYL AND L.DT. IF A SECTOR IS FOUND IN
THIS FIELD THAT IS BAD FOR THAT CYLINDER AND TRACK,

1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715

THE SECTOR NUMBER IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS IS PLACED ON THE STACK AFTER THE ENTIRE FIELD IS SEARCHED.

CALL: JSR R4 BDSRCK
<ADDRESS OF FIELD TO BE SEARCHED>

DATA GENERATION AND COMPARE ROUTINE

CALLS: JSR R4 GENCOM
CONTROL WORD

JSR R4 GENCOM
CONTROL WORD
LENGTH

JSR R4 GENCOM
CONTROL WORD
RELOCATION CONSTANT
LENGTH

RETURN: RTS R4

R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE CONTROL WORD. THIS IS UNCONDITIONAL.

THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED. IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.

THE THIRD CALL IS IDENTICAL TO THE SECOND.

DEFINITION OF CONTROL WORD:

- BIT 15 - DO COMPARE OPERATION OF Ibuff (SOURCE) TO Obuff (DESTINATION). EXPECTED VALUES ARE IN Obuff (DESTINATION).
- BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
- BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
- BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
- BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
- BIT 10 - CLEAR Ibuff TO PATTERN SELECTED.
- BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
- BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.

1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771

BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES. OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16. REFER TO THE PROGRAM LISTING FOR PAT02 THROUGH PAT16.

EXPLANATION OF CALLS:

THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.

THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED FOR DATA GENERATION OR COMPARE AND FOR Ibuff INITIALIZATION.

THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.

DESCRIPTION:

THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:

- A. BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8) OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK ARE LISTED THEREIN (BIT 9).
- B. COMPARE THE CONTENTS OF Ibuff AND Obuff (BIT 15). THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8) WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED IN L.CS1. THE HEADERS MAY BE BUILT BEFORE THE COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
- C. RESUME COMPARE OPERATION. IF A COMPARE OPERATION DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14 SET IN THE CONTROL WORD.
- D. DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT. MEMORY MANAGEMENT CAN BE INVOKED FOR

H03

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 33

SEG 0033

1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791

EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE, DATA COMPARE WILL COMPARE THE CONTENTS OF IBUFF TO ANY AREA OF AVAILABLE MEMORY.

PHASE LOCKED LOOP CLOCK ADJUSTMENT ROUTINE

THIS ROUTINE IS ENTERED VIA START LOCATION 220(8). THE PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE. THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND RESETS THE DIAGNOSTIC MODE BIT IN RKMRI. INSTRUCTIONS ON WHERE TO SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.

THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.

%

```

1792 ; *** REV 007 ***
1793
1794 .NLIST CND,MD,MC,TOC
1795 .LIST ME
1796 .ENABL ABS,AMA
1797 167400 $SWR= 167400
1798 000001 $TN= 1
1799 .TITLE CZR6KE0 RK6 FCTNL CTRL DIAG
1800 ;*COPYRIGHT (C) 1976,1977
1801 ;*DIGITAL EQUIPMENT CORP.
1802 ;*MAYNARD, MASS. 01754
1803 ;*
1804 ;*PROGRAM BY MARV TEGROTENHUIS
1805 ;*
1806 ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1807 ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
1808 ;*
1809 .SBTTL OPERATIONAL SWITCH SETTINGS
1810 ;*
1811 ;* SWITCH USE
1812 ;*-----
1813 ;* 15 HALT ON ERROR
1814 ;* 14 LOOP ON TEST
1815 ;* 13 INHIBIT ERROR TYPEOUTS
1816 ;* 12 ABORT PROGRAM AFTER 20 ERRORS
1817 ;* 11 INHIBIT ITERATIONS
1818 ;* 10 BELL ON ERROR
1819 ;* 9 LOOP ON ERROR
1820 ;* 8 LOOP ON TEST IN SWR<7:0>
1821 .SBTTL BASIC DEFINITIONS
1822
1823 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
1824 001100 STACK= 1100
1825 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
1826 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
1827
1828 ;*MISCELLANEOUS DEFINITIONS
1829 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
1830 000012 LF= 12 ;;CODE FOR LINE FEED
1831 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
1832 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
1833 177776 PS= 177776 ;;PROCESSOR STATUS WORD
1834 .EQUIV PS,PSW
1835 177774 STKLMT= 177774 ;;STACK LIMIT REGISTER
1836 177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
1837 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
1838 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
1839
1840 ;*GENERAL PURPOSE REGISTER DEFINITIONS
1841 000000 R0= %0 ;;GENERAL REGISTER
1842 000001 R1= %1 ;;GENERAL REGISTER
1843 000002 R2= %2 ;;GENERAL REGISTER
1844 000003 R3= %3 ;;GENERAL REGISTER
1845 000004 R4= %4 ;;GENERAL REGISTER
1846 000005 R5= %5 ;;GENERAL REGISTER
1847 000006 R6= %6 ;;GENERAL REGISTER

```

1848 000007
1849 000006
1850 000007

R7= %7 ;:GENERAL REGISTER
SP= %6 ;:STACK POINTER
PC= %7 ;:PROGRAM COUNTER

1851
1852
1853 000000
1854 000040
1855 000100
1856 000140
1857 000200
1858 000240
1859 000300
1860 000340

:*PRIORITY LEVEL DEFINITIONS
PR0= 0 ;:PRIORITY LEVEL 0
PR1= 40 ;:PRIORITY LEVEL 1
PR2= 100 ;:PRIORITY LEVEL 2
PR3= 140 ;:PRIORITY LEVEL 3
PR4= 200 ;:PRIORITY LEVEL 4
PR5= 240 ;:PRIORITY LEVEL 5
PR6= 300 ;:PRIORITY LEVEL 6
PR7= 340 ;:PRIORITY LEVEL 7

1861
1862
1863 100000
1864 040000
1865 020000
1866 010000
1867 004000
1868 002000
1869 001000
1870 000400
1871 000200
1872 000100
1873 000040
1874 000020
1875 000010
1876 000004
1877 000002
1878 000001

:*"SWITCH REGISTER" SWITCH DEFINITIONS
SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1

1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889

.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

1890
1891 100000
1892 040000
1893 020000
1894 010000
1895 004000
1896 002000
1897 001000
1898 000400
1899 000200
1900 000100
1901 000040
1902 000020
1903 000010

:*DATA BIT DEFINITIONS (BIT00 TO BIT15)
BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10

1904	000004	BIT02=	4
1905	000002	BIT01=	2
1906	000001	BIT00=	1
1907		.EQUIV	BIT09,BIT9
1908		.EQUIV	BIT08,BIT8
1909		.EQUIV	BIT07,BIT7
1910		.EQUIV	BIT06,BIT6
1911		.EQUIV	BIT05,BIT5
1912		.EQUIV	BIT04,BIT4
1913		.EQUIV	BIT03,BIT3
1914		.EQUIV	BIT02,BIT2
1915		.EQUIV	BIT01,BIT1
1916		.EQUIV	BIT00,BIT0

1918		;*BASIC "CPU" TRAP VECTOR ADDRESSES		
1919	000004	ERRVEC=	4	TIME OUT AND OTHER ERRORS
1920	000010	RESVEC=	10	RESERVED AND ILLEGAL INSTRUCTIONS
1921	000014	TBITVEC=	14	"T" BIT
1922	000014	TRTVEC=	14	TRACE TRAP
1923	000014	BPTVEC=	14	BREAKPOINT TRAP (BPT)
1924	000020	IOTVEC=	20	INPUT/OUTPUT TRAP (IOT) **SCOPE**
1925	000024	PWRVEC=	24	POWER FAIL
1926	000030	EMTVEC=	30	EMULATOR TRAP (EMT) **ERROR**
1927	000034	TRAPVEC=	34	"TRAP" TRAP
1928	000060	TKVEC=	60	TTY KEYBOARD VECTOR
1929	000064	TPVEC=	64	TTY PRINTER VECTOR
1930	000240	PIRQVEC=	240	PROGRAM INTERRUPT REQUEST VECTOR

.SBTTL MEMORY MANAGEMENT DEFINITIONS

;*KT11 VECTOR ADDRESS

1934 000250 MMVEC= 250

;*KT11 STATUS REGISTER ADDRESSES

1939	177572	SR0=	177572
1940	177574	SR1=	177574
1941	177576	SR2=	177576
1942	172516	SR3=	172516

;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

1946	172300	KIPDR0=	172300
1947	172302	KIPDR1=	172302
1948	172304	KIPDR2=	172304
1949	172306	KIPDR3=	172306
1950	172310	KIPDR4=	172310
1951	172312	KIPDR5=	172312
1952	172314	KIPDR6=	172314
1953	172316	KIPDR7=	172316

;*KERNEL "I" PAGE ADDRESS REGISTERS

1957	172340	KIPAR0=	172340
1958	172342	KIPAR1=	172342
1959	172344	KIPAR2=	172344

```

1960      172346      KIPAR3= 172346
1961      172350      KIPAR4= 172350
1962      172352      KIPAR5= 172352
1963      172354      KIPAR6= 172354
1964      172356      KIPAR7= 172356
1965
1966      000210      AVECT1= 210      ;DEFINE RK611 VECTOR INTERRUPT
1967      000240      APRIOR= PR5      ;DEFINE RK611 PRIORITY
1968      177440      ABASE= 177440    ;DEFINE RK611 BASE BUS ADDRESS
1969
1970      .SBTTL  RK611 CONTROLLER REGISTER DEFINITION
1971
1972      000000      RKCS1= 0      ;CONTROL AND STATUS REGISTER 1
1973      000002      RKWC= 2      ;WORD COUNT REGISTER
1974      000004      RKBA= 4      ;BUS ADDRESS REGISTER
1975      000006      RKDA= 6      ;DESIRED TRACK SECTOR REGISTER
1976      000010      RKCS2= 10     ;CONTROL AND STATUS REGISTER 2
1977      000012      RKDS= 12     ;DRIVE STATUS REGISTER
1978      000014      RKER= 14     ;ERROR REGISTER
1979      000016      RKASOF= 16    ;ATTENTION SUMMARY AND OFFSET REGISTER
1980      000020      RKDCYL= 20    ;DESIRED CYLINDER REGISTER
1981      000024      RKDB= 24     ;DATA BUFFER
1982      000026      RKMR1= 26     ;MAINTENANCE REGISTER 1
1983      000034      RKMR2= 34     ;MAINTENANCE REGISTER 2
1984      000036      RKMR3= 36     ;MAINTENANCE REGISTER 3
1985      000030      RKECPS= 30    ;ECC POSITION INFORMATION
1986      000032      RKECPT= 32    ;ECC PATTERN INFORMATION
1987      000022      RKSPAR= 22    ;SPARE REGISTER
1988
1989      .SBTTL  DRIVE COMMANDS
1990
1991      000101      SELDRV= 101    ;SELECT DRIVE
1992      000103      PACK= 103     ;PACK ACKNOWLEDGE
1993      000105      CLEAR= 105    ;DRIVE CLEAR
1994      000107      UNLOAD= 107   ;UNLOAD
1995      000111      SRTSPL= 111   ;START SPINDLE
1996      000113      RECAL= 113    ;RECALIBRATE
1997      000115      OFFSET= 115   ;OFFSET
1998      000117      SEEK= 117     ;SEEK
1999      000121      RDDATA= 121   ;READ DATA
2000      000123      WRDATA= 123   ;WRITE DATA
2001      000125      RDHEAD= 125   ;READ HEADER
2002      000127      WRHEAD= 127   ;WRITE HEADER AND DATA
2003      000131      WRTCHK= 131   ;WRITE CHECK
2004      000300      INTR= 300     ;GENERATE INTERRUPT TO CPU
2005
2006      .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
2007
2008      000001      GO= BIT0      ;GO BIT
2009      000100      IE= BIT6      ;INTERRUPT ENABLE
2010      000200      RDY= BIT7      ;CONTROLLER READY
2011      000400      BA16= BIT8     ;BUS ADDRESS BIT 16
2012      001000      BA17= BIT9     ;BUS ADDRESS BIT 17
2013      002000      CDT= BIT10    ;CONTROLLER DRIVE TYPE (0=RK06)
2014      004000      CTO= BIT11    ;CONTROLLER TIMED OUT WAITING FOR
2015      ; DRIVE RESPONSE

```

M03

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 38
CONTROL AND STATUS REGISTER 1 BITS

SEQ 0038

2016	010000	CFMT=	BIT12	; CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
2017	020000	SPAR=	BIT13	; DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
2018	040000	DI=	BIT14	; DRIVE INTERRUPT
2019	100000	CERR=	BIT15	; CONTROLLER ERROR
2020	100000	CCLR=	BIT15	; CONTROLLER CLEAR

.SBTTL CONTROL AND STATUS REGISTER 2 BITS

2024	000007	DRVMSK=	7	; MASK FOR DRIVE SELECTION CODE
2025	000010	RLS=	BIT3	; DESELECT OR RELEASE DRIVE IN BITS 0-2
2026	000020	BAI=	BIT4	; BUS ADDRESS INCREMENT INHIBIT
2027	000040	SCLR=	BIT5	; CLEAR CONTROLLER AND ALL DRIVES
2028	000100	IR=	BIT6	; INPUT READY
2029	000200	OR=	BIT7	; OUTPUT READY
2030	000400	UFE=	BIT8	; UNIT FIELD ERROR
2031	001000	MDS=	BIT9	; MULTIPLE DRIVE SELECT
2032	002000	PGE=	BIT10	; PROGRAMMING ERROR
2033	004000	NEM=	BIT11	; NON-EXISTENT MEMORY
2034	010000	NED=	BIT12	; NON-EXISTENT DRIVE
2035	020000	UPE=	BIT13	; UNIBUS PARITY ERROR
2036	040000	WCE=	BIT14	; WRITE CHECK ERROR
2037	100000	DLT=	BIT15	; DATA LATE ERROR

.SBTTL ERROR REGISTER BIT DEFINITION

2041	000001	ILF=	BIT0	; ILLEGAL FUNCTION CODE
2042	000002	SKI=	BIT1	; SEEK INCOMPLETE
2043	000004	NXF=	BIT2	; NON-EXECUTABLE DRIVE FUNCTION
2044	000010	DRPAR=	BIT3	; DRIVE DETECTED DRIVE BUS PARITY ERROR
2045	000020	FMTE=	BIT4	; FORMAT ERROR
2046	000040	DTYPE=	BIT5	; DRIVE TYPE ERROR
2047	000100	ECH=	BIT6	; ECC HARD
2048	000200	BSE=	BIT7	; BAD SECTOR ERROR
2049	000400	HVRC=	BIT8	; HEADER VRC ERROR
2050	001000	COE=	BIT9	; CYLINDER ADDRESS OVERFLOW ERROR
2051	002000	IDAE=	BIT10	; INVALID DISK ADDRESS ERROR
2052	004000	WLE=	BIT11	; WRITE LOCK ERROR
2053	010000	DTE=	BIT12	; DRIVE TIMING ERROR
2054	020000	OPI=	BIT13	; OPERATION (SEARCH) INCOMPLETE
2055	040000	UNS=	BIT14	; DRIVE UNSAFE
2056	100000	DCK=	BIT15	; DATA CHECK

.SBTTL STATUS REGISTER BIT DEFINITION

2060	000001	DRA=	BIT0	; DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
2062	000004	OFST=	BIT2	; DRIVE OFFSET
2063	000010	ACLO=	BIT3	; AC LOW
2064	000020	SPDLSS=	BIT4	; SPEED LOSS
2065	000040	DROT=	BIT5	; DRIVE OFF TRACK
2066	000100	VV=	BIT6	; VOLUME VALID
2067	000200	DRDY=	BIT7	; DRIVE READY
2068	000400	DDT=	BIT8	; DRIVE TYPE (0=RK06)
2069	004000	WRL=	BIT11	; WRITE LOCK
2070	020000	PIP=	BIT13	; POSITIONING IN PROGRESS
2071	040000	DSC=	BIT14	; DRIVE STATUS CHANGE


```

2072      100000      SVAL=   BIT15      ;STATUS VALID
2073
2074      .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION
2075
2076      000017      MESMSK= 17      ;MESSAGE MASK
2077
2078      000020      PAT=     BIT4      ;FORCE EVEN PARITY ON SERCON MESSAGE LINES
2079      000040      DMD=     BIT5      ;DIAGNOSTIC MODE
2080      000100      MSP=     BIT6      ;MAINTENANCE SECTOR PULSE
2081      000200      MIND=    BIT7      ;MAINTENANCE INDEX
2082      000400      MCLK=    BIT8      ;MAINTENANCE CLOCK
2083      001000      MERD=    BIT9      ;MAINTENANCE ENCODED READ DATA
2084      002000      MEWD=    BIT10     ;MAINTENANCE ENCODED WRITE DATA
2085      004000      PCA=     BIT11     ;PRECOMPENSATION ADVANCE
2086      010000      PCD=     BIT12     ;PRECOMPENSATION DELAY
2087      020000      ECCW=    BIT13     ;ECC WORD IS BEING READ OR WRITTEN
2088      040000      WRTGAT= BIT14     ;WRITE GATE
2089      100000      RDGATE= BIT15     ;READ GATE
2090
2091      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
2092
2093      000040      S.DRA=   BIT5      ;DRIVE AVAILIABLE
2094      000100      S.VV=   BIT6      ;VOLUME VALID
2095      000200      S.DRY=   BIT7      ;DRIVE READY
2096      000400      S.TYPE=  BIT8      ;DRIVE TYPE
2097      001000      S.FORM=  BIT9      ;DRIVE FORMAT
2098      002000      S.OFF=  BIT10     ;OFFSET
2099      004000      S.WRL=  BIT11     ;WRITE LOCK
2100      010000      S.SPIN=  BIT12     ;SPINDLE ON
2101      020000      S.PIP=  BIT13     ;POSITIONING IN PROGRESS
2102      040000      S.DSC=  BIT14     ;DRIVE STATUS CHANGE
2103
2104      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
2105
2106      000040      S.ICYL=  BIT5      ;ILLEGAL CYLINDER ADDRESS
2107      000100      S.ACLO=  BIT6      ;AC LOW
2108      000200      S.FLT=  BIT7      ;DRIVE FAULT
2109      000400      S.ILF=  BIT8      ;ILLEGAL FUNCTION
2110      001000      S.PAR=  BIT9      ;DRIVE DETECTED SERCON PARITY ERROR
2111      002000      S.SKI=  BIT10     ;SEEK INCOMPLETE
2112      004000      S.WLE=  BIT11     ;WRITE LOCK ERROR
2113      010000      S.SPLS= BIT12     ;SPEED LOSS
2114      020000      S.DROT= BIT13     ;DRIVE OFF TRACK
2115      040000      S.UNS=  BIT14     ;DRIVE UNSAFE
2116
2117      .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
2118
2119      000020      S.XDOK=  BIT4      ;TRANSDUCER OK
2120      000040      S.HDHM=  BIT5      ;HEADS HOME
2121      000100      S.BRHM=  BIT6      ;BRUSHES HOME
2122      000200      S.DOOR=  BIT7      ;DOOR INTERLOCKED
2123      000400      S.CART=  BIT8      ;CARTRAGE INTERLOCK
2124      001000      S.SPOK=  BIT9      ;SPEED OK
2125      002000      S.FWD=  BIT10     ;FORWARD
2126      004000      S.REV=  BIT11     ;REVERSE
2127      010000      S.LOAD=  BIT12     ;HEADS LOADING

```

```

2128          020000          S.RTZ= BIT13          ;RETURN TO ZERO
2129          040000          S.UNLD= BIT14          ;HEADS UNLOADING
2130
2131          .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
2132
2133          000020          S.SECT= BIT4          ;SECTOR ERROR
2134          000040          S.WCLK= BIT5          ;WRITE CLOCK AND NO WRITE GATE
2135          000100          S.WGAT= BIT6          ;WRITE GATE AND NO TRANSISTIONS
2136          000200          S.HDFL= BIT7          ;HEAD FAULT
2137          000400          S.MHD= BIT8          ;MULTIPLE HEAD SELECT
2138          001000          S.XERR= BIT9          ;INDEX ERROR
2139          002000          S.DIB= BIT10         ;DIBIT ERROR
2140          004000          S.PLO= BIT11         ;PLO ERROR
2141          010000          S.NMOV= BIT12         ;SEEK AND NO MOTION
2142          020000          S.LIMD= BIT13         ;LIMIT DETECT ON SEEK
2143          040000          S.BRKE= BIT14         ;SERVO-BRAKE
2144
2145          .SBTTL  COMMON MASKS
2146
2147          000007          M.DRV= 7          ;DRIVE CODE
2148          100000          M.PAR= BIT15         ;PARITY
2149          000003          M.ID= 3          ;BYTE ID
2150          017760          M.CDIF= 17760        ;CYLINDER DIFFERENCE/OFFSET
2151          017760          M.CADD= 17760        ;CYLINDER ADDRESS
2152          077770          M.SER= 77770        ;DRIVE SERIAL NUMBER
2153          000760          M.SECT= 760         ;SECTOR COUNT
2154          007000          M.HEAD= 7000        ;HEAD DECODE
2155
2156          .SBTTL  TRAP CATCHER
2157          000000          .=0
2158          ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
2159          ;*A "+2, IOT" SEQUENCE TO CATCH AND PROCESS ILLEGAL
2160          ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
2161          ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
2162          ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
2163          ;*LESS THAN 1002) JUMPS TO THE $ERROR ROUTINE.
2164          ;*THE $ERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
2165          ;* PC=YYYYYY UNEXPECTED TRAP TO XXX
2166          ;*AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
2167          ;*WHERE XXX=LOCATION OF ILLEGAL TRAP
2168          ;* YYYYYY=PC AT TIME OF TRAP
2169          ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
2170          ;* CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.
2171
2172          000000  000000          $4OCAT: HALT          ;:HALT
2173          000002  000737          BR          .-100          ;:BRANCH TO 177700 & TIME OUT (NOT ON
2174          ;:11/05)
2175          000004  001772          .WORD  START          ;:VECTOR TO STARTING ADDRESS
2176          000006  000340          .WORD  340          ;:WITH PRIORITY LEVEL 7
2177          000174  000174          .=174
2178          000174  000000          DISPREG: .WORD  0          ;:SOFTWARE DISPLAY REGISTER
2179          000176  000000          SWREG: .WORD  0          ;:SOFTWARE SWITCH REGISTER
2180          .SBTTL  STARTING ADDRES(ES)
2181          000200  000137  001772          JMP  @#START ;;GO TO START OF PROGRAM
2182          000204  000137  001762          JMP  RESTRT          ;:JUMP TO RESTART ROUTINE
2183          000214

```

2184	000214	000137	001752
2185		000220	
2186	000220	000137	001742
2187			
2188			
2189			
2190			
2191			
2192		000224	
2193		000024	
2194	000024	000200	
2195		000044	
2196	000044	000224	
2197		000224	
2198			
2199			
2200			
2201			
2202	000224		
2203	000224	000000	
2204	000226	001276	
2205	000230	000024	
2206	000232	000074	
2207	000234	000740	
2208	000236	000031	

```

      JMP      PARM          ;JUMP TO OPERATOR ASSIGNED PARAMETERS
      .=220
      JMP      SETCLK       ;JUMP TO SET CLOCK ROUTINE
.SBTTL  APT PARAMETER BLOCK

```

```

;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
      .SX=.          ;SAVE CURRENT LOCATION
      .=24          ;SET POWER FAIL TO POINT TO START OF PROGRAM
      200          ;FOR APT START UP
      .=44          ;POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR      ;POINT TO APT HEADER BLOCK
      .=$X          ;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

```

```

$APTHD:
$HIBTS: .WORD 0          ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL     ;ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT:  .WORD 20        ;RUN TIM OF LONGEST TEST
$PASTM: .WORD 60        ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 480       ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

```

2209
2210
2211
2212
2213
2214
2215
2216 001100 001100
2217 001100 000000
2218 001102 000
2219 001103 000
2220 001104 000000
2221 001106 000000
2222 001110 000000
2223 001112 000000
2224 001114 000
2225 001115 001
2226 001116 000000
2227 001120 000000
2228 001122 000000
2229 001124 000000
2230 001126 000000
2231 001130 000000
2232 001132 000000
2233 001134 000
2234 001135 000
2235 001136 000000
2236 001140 177570
2237 001142 177570
2238 001144 177560
2239 001146 177562
2240 001150 177564
2241 001152 177566
2242 001154 000
2243 001155 002
2244 001156 012
2245 001157 000
2246 001160 000000
2247
2248 001162 000000
2249 001164 000000
2250 001166 000000
2251 001170 000000
2252 001172 000000
2253 001174 000000
2254 001176 000000
2255 001200 000000
2256 001202 000000
2257 001204 000000
2258 001206 000000
2259 001210 000000
2260 001212 000000
2261 001214 000000
2262 001216 000000
2263 001220 000000
2264 001222 000000

```

.SBTTL COMMON TAGS

```

*****
*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.

```

SCMTAG: .=1100

```

$STNM: .WORD 0
$ERFLG: .BYTE 0
$ICNT: .WORD 0
$LPADR: .WORD 0
$LPERR: .WORD 0
$ERTTL: .WORD 0
$ITEMB: .BYTE 0
$ERMAX: .BYTE 1
$ERRPC: .WORD 0
$GDADR: .WORD 0
$BDADR: .WORD 0
$GDDAT: .WORD 0
$BDDAT: .WORD 0
$AUTOB: .BYTE 0
$INTAG: .BYTE 0
$SWR: .WORD DSWR
$DISPLAY: .WORD DDISP
$TKS: 177560
$TKB: 177562
$TPS: 177564
$TPB: 177566
$NULL: .BYTE 0
$FILLS: .BYTE 2
$FILLC: .BYTE 12
$TPFLG: .BYTE 0
$REGAD: .WORD 0
$REG0: .WORD 0
$REG1: .WORD 0
$REG2: .WORD 0
$REG3: .WORD 0
$REG4: .WORD 0
$REG5: .WORD 0
$REG6: .WORD 0
$REG7: .WORD 0
$REG10: .WORD 0
$REG11: .WORD 0
$REG12: .WORD 0
$REG13: .WORD 0
$REG14: .WORD 0
$REG15: .WORD 0
$REG16: .WORD 0
$REG17: .WORD 0
$TMPO: .WORD 0

```

;; START OF COMMON TAGS

```

;; CONTAINS THE TEST NUMBER
;; CONTAINS ERROR FLAG
;; CONTAINS SUBTEST ITERATION COUNT
;; CONTAINS SCOPE LOOP ADDRESS
;; CONTAINS SCOPE RETURN FOR ERRORS
;; CONTAINS TOTAL ERRORS DETECTED
;; CONTAINS ITEM CONTROL BYTE
;; CONTAINS MAX. ERRORS PER TEST
;; CONTAINS PC OF LAST ERROR INSTRUCTION
;; CONTAINS ADDRESS OF 'GOOD' DATA
;; CONTAINS ADDRESS OF 'BAD' DATA
;; CONTAINS 'GOOD' DATA
;; CONTAINS 'BAD' DATA
;; RESERVED--NOT TO BE USED
;; AUTOMATIC MODE INDICATOR
;; INTERRUPT MODE INDICATOR
;; ADDRESS OF SWITCH REGISTER
;; ADDRESS OF DISPLAY REGISTER
;; TTY KBD STATUS
;; TTY KBD BUFFER
;; TTY PRINTER STATUS REG. ADDRESS
;; TTY PRINTER BUFFER REG. ADDRESS
;; CONTAINS NULL CHARACTER FOR FILLS
;; CONTAINS # OF FILLER CHARACTERS REQUIRED
;; INSERT FILL CHARS. AFTER A "LINE FEED"
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
;; CONTAINS THE ADDRESS FROM
;; WHICH ($REG0) WAS OBTAINED
;; CONTAINS (($REGAD)+0)
;; CONTAINS (($REGAD)+2)
;; CONTAINS (($REGAD)+4)
;; CONTAINS (($REGAD)+6)
;; CONTAINS (($REGAD)+10)
;; CONTAINS (($REGAD)+12)
;; CONTAINS (($REGAD)+14)
;; CONTAINS (($REGAD)+16)
;; CONTAINS (($REGAD)+20)
;; CONTAINS (($REGAD)+22)
;; CONTAINS (($REGAD)+24)
;; CONTAINS (($REGAD)+26)
;; CONTAINS (($REGAD)+30)
;; CONTAINS (($REGAD)+32)
;; CONTAINS (($REGAD)+34)
;; CONTAINS (($REGAD)+36)
;; USER DEFINED

```

```

2265 001224 000000 $TMP1: .WORD 0 ;; USER DEFINED
2266 001226 000000 $TMP2: .WORD 0 ;; USER DEFINED
2267 001230 000000 $TMP3: .WORD 0 ;; USER DEFINED
2268 001232 000000 $TMP4: .WORD 0 ;; USER DEFINED
2269 001234 000000 $TMP5: .WORD 0 ;; USER DEFINED
2270 001236 000000 $TMP6: .WORD 0 ;; USER DEFINED
2271 001240 000000 $TMP7: .WORD 0 ;; USER DEFINED
2272 001242 000000 $TMP10: .WORD 0 ;; USER DEFINED
2273 001244 000000 $TMP11: .WORD 0 ;; USER DEFINED
2274 001246 000000 $TMP12: .WORD 0 ;; USER DEFINED
2275 001250 000000 $TMP13: .WORD 0 ;; USER DEFINED
2276 001252 000000 $TMP14: .WORD 0 ;; USER DEFINED
2277 001254 000000 $TMP15: .WORD 0 ;; USER DEFINED
2278 001256 000000 $TMP16: .WORD 0 ;; USER DEFINED
2279 001260 000000 $TMP17: .WORD 0 ;; USER DEFINED
2280 001262 000000 $TIMES: 0 ;; MAX. NUMBER OF ITERATIONS
2281 001264 000000 $ESCAPE: 0 ;; ESCAPE ON ERROR ADDRESS
2282 001266 177607 000377 $BELL: .ASCIZ <207><377><377> ;; CODE FOR BELL
2283 001272 077 $QUES: .ASCII /?/ ;; QUESTION MARK
2284 001273 015 $CRLF: .ASCII <15> ;; CARRIAGE RETURN
2285 001274 000012 $LF: .ASCIZ <12> ;; LINE FEED
2286 ;; *****
2287 .SBTTL APT MAILBOX-ETABLE
2288
2289 ;; *****
2290 .EVEN
2291 001276 $MAIL: ;; APT MAILBOX
2292 001276 000000 $MSGTY: .WORD AMSGTY ;; MESSAGE TYPE CODE
2293 001300 000000 $FATAL: .WORD AFATAL ;; FATAL ERROR NUMBER
2294 001302 000000 $TESTN: .WORD ATESTN ;; TEST NUMBER
2295 001304 000000 $PASS: .WORD APASS ;; PASS COUNT
2296 001306 000000 $DEVCT: .WORD ADEVCT ;; DEVICE COUNT
2297 001310 000000 $UNIT: .WORD AUNIT ;; I/O UNIT NUMBER
2298 001312 000000 $MSGAD: .WORD AMSGAD ;; MESSAGE ADDRESS
2299 001314 000000 $MSGLG: .WORD AMSGLG ;; MESSAGE LENGTH
2300 001316 $ETABLE: ;; APT ENVIRONMENT TABLE
2301 001316 000 $ENV: .BYTE AENV ;; ENVIRONMENT BYTE
2302 001317 000 $ENVM: .BYTE AENVM ;; ENVIRONMENT MODE BITS
2303 001320 000000 $SWREG: .WORD ASWREG ;; APT SWITCH REGISTER
2304 001322 000000 $USWR: .WORD AUSWR ;; USER SWITCHES
2305 001324 000000 $CPUOP: .WORD ACPUOP ;; CPU TYPE, OPTIONS
2306 ;; BIT 15-11=CPU TYPE
2307 ;; 11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
2308 ;; 11/70=06, PDQ=07, Q=10
2309 ;; BIT 10=REAL TIME CLOCK
2310 ;; BIT 9=FLOATING POINT PROCESSOR
2311 ;; BIT 8=MEMORY MANAGEMENT
2312 001326 000 $MAMS1: .BYTE AMAMS1 ;; HIGH ADDRESS, M.S. BYTE
2313 001327 000 $MTYP1: .BYTE AMTYP1 ;; MEM. TYPE, BLK#1
2314 ;; MEM. TYPE BYTE -- (HIGH BYTE)
2315 ;; 900 NSEC CORE=001
2316 ;; 300 NSEC BIPOLAR=002
2317 ;; 500 NSEC MOS=003
2318 001330 000000 $MADR1: .WORD AMADR1 ;; HIGH ADDRESS, BLK#1
2319 ;; MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
2320 001332 000 $MAMS2: .BYTE AMAMS2 ;; HIGH ADDRESS, M.S. BYTE

```

2321	001333	000	\$MTYP2:	.BYTE	AMTYP2	::	MEM.TYPE, BLK#2
2322	001334	000000	\$MADR2:	.WORD	AMADR2	::	MEM.LAST ADDRESS, BLK#2
2323	001336	000	\$MAMS3:	.BYTE	AMAMS3	::	HIGH ADDRESS, M.S.BYTE
2324	001337	000	\$MTYP3:	.BYTE	AMTYP3	::	MEM.TYPE, BLK#3
2325	001340	000000	\$MADR3:	.WORD	AMADR3	::	MEM.LAST ADDRESS, BLK#3
2326	001342	000	\$MAMS4:	.BYTE	AMAMS4	::	HIGH ADDRESS, M.S.BYTE
2327	001343	000	\$MTYP4:	.BYTE	AMTYP4	::	MEM.TYPE, BLK#4
2328	001344	000000	\$MADR4:	.WORD	AMADR4	::	MEM.LAST ADDRESS, BLK#4
2329	001346	000210	\$VECT1:	.WORD	AVECT1	::	INTERRUPT VECTOR#1, BUS PRIORITY#1
2330	001350	000000	\$VECT2:	.WORD	AVECT2	::	INTERRUPT VECTOR#2, BUS PRIORITY#2
2331	001352	177440	\$BASE:	.WORD	ABASE	::	BASE ADDRESS OF EQUIPMENT UNDER TEST
2332	001354	000000	\$DEVN:	.WORD	ADEVN	::	DEVICE MAP
2333	001356	000000	\$CDW1:	.WORD	ACDW1	::	CONTROLLER DESCRIPTION WORD#1
2334	001360		\$ETEND:				
2335			.MEXIT				

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ::POINTS TO THE ERROR MESSAGE
;* DH ::POINTS TO THE DATA HEADER
;* DT ::POINTS TO THE DATA
;* DF ::POINTS TO THE DATA FORMAT

001360

\$ERRTB:

;* EM AND DH ARE ASCIZ DATA. EM IS ALWAYS A MESSAGE BUT DH
;* CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACCROSS
;* THE PAGE, DT IS A STRING OF WORDS THAT POINT TO THE DATA TO
;* BE TYPED, AND DF IS A STRING OF WORD THAT TELL HOW THE DT WORDS
;* ARE TO BE TYPED. IF ANY OF THE POINTERS ARE NOT NEEDED, FOR A
;* PARTICULAR FORMAT, IT IS REPLACED WITH A ZERO.
;* THE NORMAL USAGE OF THE ERROR TABLE IS TO HAVE A TABLE ENTRY FOR
;* EACH ERROR MESSAGE THAT CAN OCCUR. IN THE INTEREST OF ECONOMICS
;* OF CORE MEMORY, THIS PROGRAM USES THE ERROR TABLE IN A
;* SLIGHTLY DIFFERENT MANNERS AS DESCRIBED BELOW.
;* THE ERROR TABLE ENTRIES MAKE UP A SET OF REPORT FORMATS THAT ARE USED
;* THROUGHOUT THE PROGRAM. WHEN AN ERROR IS TO BE REPORTED, THE
;* TABLE ENTRY THAT PROVIDES THE DESIRED FORMAT IS CHOSEN FROM
;* THE DEFINED SET. THE TABLE ENTRY CHOSEN IS THEN ALTERED
;* BY CHANGING THE FIRST (AND POSSIBLY THE SECOND) WORD TO CONTAIN
;* THE ADDRESS OF THE ASCIZ STRING THAT MAKES UP THE MESSAGE
;* PORTION OF THE REPORT. THE DATA FIELDS FOR THAT ENTRY ARE NEVER
;* CHANGED, NOR ARE THE COLUMN LABELS OR POSITIONS.
;* THE FORMAT THAT EACH TABLE ENTRY PROVIDES IS SHOWN BELOW WITH
;* THE DEFINITION OF THE ENTRY. ALL DATA FIELDS ARE TYPED IN OCTAL.

;* :ERROR ITEM 1
;* (MESSAGE)
;* TST NUM ERR PC DRIVE
;* \$TESTN \$ERRPC DRVNUM

EMIN: .WORD 0
DH001
DT001
DF001

;* :ERROR ITEM 2
;* (MESSAGE)
;* (MESSAGE)
;* TST NUM ERR PC DRIVE
;* \$TESTN \$ERRPC DRVNUM
;* RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA

2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391

001360 000000
001362 057600
001364 061152
001366 061244

ERROR POINTER TABLE

2392
2393
2394
2395
2396 001370 000000
2397 001372 000000
2398 001374 061160
2399 001376 061250
2400
2401
2402
2403
2404
2405
2406
2407 001400 000000
2408 001402 057626
2409 001404 061130
2410 001406 061270
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447

```

:*      T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.DCYL  T.DA
:*      RKBA   RKWC
:*      T.BA   T.WC
EM2N:  .WORD  0
DH2N:  .WORD  0
      DT002
      DF002
:*      *ERROR ITEM 3
      (MESSAGE)
:*      TST NUM ERR PC  DRIVE
:*      $TESTN $ERRPC  DRVNUM
:*      RKCS1  RKCS2  RKDS  RKER  RKASOF  RKMRI
:*      T.CS1  T.CS2  T.DS  T.ER  T.AST  T.MRI

EM3N:  .WORD  0
DH002A DT003
      DF003

```

```

:* ERROR ITEMS 4,5,6,&7 ARE USED TO REPORT ERRORS THAT ARE THE RESULT
:* OF A HARDWARE ERROR INDICATOR BEING SET WHEN NOT EXPECTED,
:* NOT SET WHEN IT IS EXPECTED, OR BOTH. THE ERROR REPORT WILL
:* CONTAIN (1) ALL THE ERRORS THAT WERE DETECTED, (2) ALL THE EXPECTED
:* ERRORS THAT DID NOT OCCUR, OR (3) ALL THE EXPEDTED BUT NOT SET ERRORS
:* AND THE UNEXPECTED BUT SET ERRORS.
:*
:* THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT.
:* INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING
:* STATEMENTS:
:*
:* "THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
:* "THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
:* "THE ABOVE ARE ERRORS SET IN OPERATION:"
:*
:* PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT
:* SPECIFY TJE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE
:* THAT IDENTIFIES THE OPERATION BEING PERFORMED.
:*
:* EXAMPLE:
:* NON-EXISTANT DRIVE
:* THE ABOVE ARE ERRORS SET IN OPERATION:
:* DRIVE SELECT
:* (ADDITIONAL LINES OF INFORMATION)
:*
:* THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.
:*
:* EXAMPLE:
:* NON-EXISTANT DRIVE
:* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
:* DRIVE SELECT
:* (ADDITIONAL LINES OF INFORMATION)
:*
:* THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR, I.E.
:* A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.

```


2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503

```

* EXAMPLE:
* NON-EXISTANT MEMORY
* THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
* UNIBUS PARITY ERROR
* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
* WRITE DATA
* (ADDITIONAL LINES OF INFORMATION)
*
* THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN UNIBUS
* PARITY ERROR WAS EXPECTED.
*
* ERROR ITEM 4
* (DESCRIPTION OF ERROR)
* ERROR IN OPERATION
* (DESCRIPTION OF OPERATION)
* TST NUM ERR PC DRIVE
* $TESTN $ERRPC $DRVNUM
* RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
* T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
* RKBA RKWC
* T.BA T.WA
* A00 B00 A01 B01 A02 B02 A03 B03
* $REG10 $REG11 $REG12 $REG13 $REG14 $REG15 $REG16 $REG17
*
* THE ERRORS REPORTED BY THIS FORMAT ARE:
* CONTROLLER DETECTED DRIVE BUS ERROR
* DRIVE DETECTED DRIVE BUS ERROR
* SEEK INCOMPLETE
* NON-EXECUTABLE DRIVE FUNCTION
* DRIVE TIMING ERROR
* DRIVE UNSAFE
* AC LOW
* SPINDLE SPEED LOSS
* DRIVE OFF TRACK
* ILLEGAL DRIVE ADDRESS ERROR
* CYLINDER OVERFLOW
* DRIVE TYPE ERROR
* FORMAT ERROR
* WRITE LOCK ERROR

```

```

001410 000000
001412 000000
001414 061160
001416 061300

```

```

EM4N: .WORD 0
DH4N: .WORD 0
DT004
DF004

```

```

* ERROR ITEM 5
* THIS ENTRY IS THE SAME AS ITEM 4 WITH THE ADDITION
* OF A MESSAGE THAT FOLLOWS. THIS MESSAGE IS:
*
* "ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID"
*
* THIS REPORT WILL BE PRINTED WHEN THE GATHERING OF DATA FOR
* A00 THRU B03 IS NOT ACCOMPLISHED WITHOUT ERROR.

```

```

001420 000000
001422 000000

```

```

EM5N: .WORD 0
DH5N: .WORD 0

```

2504 001424 061160
2505 001426 061330

DT005
DF005

:*
:* ERROR ITEM 6
:* (DESCRIPTION OF ERROR)
:* ERROR IN OPERATION
:* (DESCRIPTION OF OPERATION)
:* TST NUM ERR PC DRIVE
:* \$TESTN \$ERRPC DRVNUM
:* RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA
:* T.CS1 T.CS2 T.DS T.ER T.ASOF T.DCYL T.DA
:* RKBA RKWC
:* T.BA T.WC

THE ERRORS REPORTED BY THIS FORMAT ARE:
DATA CHECK
WRITE CHECK
ECC HARD
DATA LATE
OPERATION INCOMPLETE
HEADER VRC ERROR
BAD SECTOR ERROR

2527 001430 000000
2528 001432 000000
2529 001434 061160
2530 001436 061364

EM6N: .WORD 0
DH6N: .WORD 0
DT006
DF006

:*
:* ERROR ITEM 7
:* (DESCRIPTION OF ERROR)
:* ERROR IN OPERATION
:* (DESCRIPTION OF OPERATION)
:* TST NUM ERR PC DRIVE
:* \$TESTN \$ERRPC DRVNUM
:* RKCS1 RKCS2 RKDS RKER RKASOF
:* T.CS1 T.CS2 T.DS T.ER T.ASOF

THE ERRORS REPORTED BY THIS FORMAT ARE:
NON-EXISTANT DRIVE
NON-EXISTANT MEMORY
CONTROLLER TIME OUT
UNIT FIELD ERROR
MULTIPLE DRIVE SELECT
PROGRAMMING ERROR
UNIBUS PARITY ERROR
ILLEGAL FUNCTION CODE
DESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL

2553 001440 000000
2554 001442 000000
2555 001444 061160
2556 001446 061410

EM7N: .WORD 0
DH7N: .WORD 0
DT007
DF007

:*
:* ERROR ITEM 10
:* (DESCRIPTION OF ERROR)

```

2560      *      ERROR AT COMPLETION OF OPERATION
2561      *      (DESCRIPITON OF OPERATION)
2562      *      TST NUM ERR PC  DRIVE
2563      *      $TESTN  $ERRPC  DRVNUM
2564      *      EXPT      IS
2565      *      $REGIO  $REGI1
2566
2567      *      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
2568      *      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
2569      *      ERRORS ARE:
2570      *      WORD COUNT INCORRECT
2571      *      BUS ADDRESS INCORRECT
2572      *      CYLINDER ADDRESS INCORRECT
2573      *      TRACK ADDRESS INCORRECT
2574      *      SECTOR ADDRESS INCORRECT
2575
2576      *      EM10N:  .WORD  0
2577      *      DHD10
2578      *      DTO15
2579      *      DFO10
2580
2581      *      ERROR ITEM 11
2582      *      (ERROR INDICATOR OR STATUS BIT)
2583      *      NOT SET AS A RESULT OF
2584      *      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
2585      *      TST NUM ERR PC  DRIVE
2586      *      $TESTN  $ERRPC  DRVNUM
2587      *      RKCS1  RKCS2  RKDS  RKER  RKASOF  RKMR1
2588      *      T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.MRI
2589
2590      *      EM11N:  .WORD  0
2591      *      DHD11
2592      *      DTO10
2593      *      DFO11
2594
2595      *      ERROR ITEM 12
2596      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2597      *      "NOT RESET AS A RESULT OF"
2598
2599      *      EM12N:  .WORD  0
2600      *      DHD12
2601      *      DTO10
2602      *      DFO11
2603
2604      *      ERROR ITEM 13
2605      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2606      *      "SET AS A RESULT OF"
2607
2608      *      EM13N:  .WORD  0
2609      *      DHD13
2610      *      DTO10
2611      *      DFO11
2612
2613      *      ERROR ITEM 14
2614      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2615      *      "RESET AS A RESULT OF"

```

C
C

2616	001510	000000
2617	001512	060670
2618	001514	061160
2619	001516	061450
2620		
2621		
2622		
2623		
2624		
2625		
2626		
2627		
2628	001520	000000
2629	001522	057626
2630	001524	061230
2631	001526	061472
2632		
2633		
2634		
2635		
2636	001530	000000
2637	001532	000000
2638	001534	061236
2639	001536	061502
2640		

```

EM14N: .WORD 0
        DH014
        DT010
        DF011

:*      ERROR ITEM 15
:*      (HEADER WORD MISCOMPARE) OR (DATA MISCOMPARE)
:*      TST NUM ERR PC  DRIVE
:*      $TESTN $ERRPC  DRVNUM
:*      GOOD   BAD    WORD NUM
:*      $REG10 $REG11 $REG12

EM15N: .WORD 0
        DH002A
        DT015
        DF015

:*      ERROR ITEM 16
:*      ADDITIONAL LINES OF GOOD, BAD, WORD NUM FOR ERROR 15

0
0
DT015A
DF016

```

2641
2642
2643
2644 001540 000000
2645 001542 000000
2646 001544 000000
2647 001546 000000
2648 001550 000000
2649 001552 000000
2650 001554 000000
2651 001556 000000
2652 001560 000000
2653 001562 000000
2654 001564 000000
2655 001566 000000
2656 001570 000000
2657 001572 000000
2658 001574 000000
2659 001576 000000
2660
2661 001600 000100
2662 001602 000000
2663 001604 000000
2664 001606
2665 001606 000
2666 001607 000
2667 001610 000000
2668 001612 000000
2669 001614 000000
2670 001616 000000
2671
2672
2673 001620 000000
2674 001622 000000
2675 001624 000000
2676
2677
2678
2679 001626 000000
2680 001630 000000
2681 001632 000024
2682
2683 001634 000024
2684 001636 061632
2685
2686 001640 061506
2687
2688 001642 000000
2689
2690 001644 000000
2691
2692 001646 000000
2693 001650 000000
2694 001652 000764
2695 001654 000017
2696 001660

.SBTTL REGISTER STORAGE FOR TEST
T.CS1: .WORD 0
T.WC: .WORD 0
T.BA: .WORD 0
T.DA: .WORD 0
T.CS2: .WORD 0
T.DS: .WORD 0
T.ER: .WORD 0
T.ASOF: .WORD 0
T.DCYL: .WORD 0
T.SPAR: .WORD 0
T.DB: .WORD 0
T.MR1: .WORD 0
T.ECPS: .WORD 0
T.ECPT: .WORD 0
T.MR2: .WORD 0
T.MR3: .WORD 0

.SBTTL REGISTER SETUP STORAGE ;PRESET WITH INTERRUPT ENABLE
L.CS1: .WORD 100
L.WC: .WORD 0
L.BA: .WORD 0
L.DA: .WORD 0
L.DS: .BYTE 0
L.DT: .BYTE 0
L.CS2: .WORD 0
L.ASOF: .WORD 0
L.DCYL: .WORD 0
L.MR1: .WORD 0

.SBTTL PROGRAM DEFINED VARIABLES

RKVEC: .WORD 0 ;RK VECTOR
RKPRI: .WORD 0 ;RK PRIORITY
SRTFLG: .WORD 0 ;START FLAG
 0 = 200
 1 = 214
 -1 = 204
DRVNUM: .WORD 0 ;DRIVE UNDER TEST
DRVBIT: .WORD 0 ;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
ERRCNT: .WORD ↑D20 ;ERROR COUNTER TO LIMIT ERROR
 ;ERRORS REPORTED IN PROGRAM
ERRLMT: .WORD ↑D20 ;DATA COMPARE ERROR LIMIT
BSF24P: .WORD BS24 ;POINTER TO BAD SECTORS 24 SECTOR MODE
 (FACTORY)
BSF26P: .WORD BS26 ;POINTER TO BAD SECTORS 26 SECTOR MODE
 (FACTORY)
BSS24P: .WORD 0 ;POINTER TO BAD SECTORS 24 SECT MODE
 (SOFTWARE)
BSS26P: .WORD 0 ;POINTER TO BAD SECTORS 26 SECTOR MODE
 (SOFTWARE)
BS26CT: .WORD 0 ;COUNT OF BAD SECTORS 26 SECTOR MODE
BS24CT: .WORD 0 ;COUNT OF BAD SECTORS 24 SECTOR MODE
MILCNT: .WORD ↑D500 ;COUNT TO APPROXIMATE 1 MILL SEC
TIMCNT: .WORD ↑D15 ;COUNTER FOR MYTIME ROUTINE
 .=1660

2697	001660	000000			
2698	001662	000000	INTSET:	.WORD	0
2699					
2700	001664	000000	OPTFLG:	.WORD	0
2701					
2702		000001			
2703		000002	DOTST=	BIT0	
2704		000004	MEMSZB=	BIT1	
2705		000010	MEMPYB=	BIT2	
2706		000100	SRTINS=	BIT3	
2707		000200	PARPRE=	BIT6	
2708		000400	BSERPT=	BIT7	
2709		002000	FPFMT=	BIT8	
2710		004000	CP1170=	BIT10	
2711		100000	DRVPT=	BIT11	
2712			LCLKPR=	BIT15	
2713	001666	000000	DRVDRP:	.WORD	0
2714	001670	000000	MEMPAR:	.WORD	0
2715	001672	000000	CSRPTR:	.WORD	0
2716					
2717	001674	000000	LCLKTK:	.WORD	0
2718	001676	000000	REFMT:	.WORD	0
2719					
2720					
2721			:		
2722	001700	000000	DESCHLD:	.WORD	0
2723	001702	000000	SPCHLD:	.WORD	0
2724	001704	000000	WRDNUM:	.WORD	0
2725	001706	000000	WRDCNT:	.WORD	0
2726	001710	177546	KWLADD:	.WORD	177546
2727	001712	000100	KWLVEC:	.WORD	100
2728	001714	172100	MEMBAS:	.WORD	172100
2729	001716	000114	MMVECA:	.WORD	114
2730					
2731	001720	000000	DTYPE:		0
2732	001722	000000	TYPTBL:		00
2733	001724	000000			00
2734	001726	000000			00
2735	001730	000000			00
2736	001732	000000			00
2737	001734	000000			00
2738	001736	000000			00
2739	001740	000000			00
2740					0

```

;NON-ZERO IF RK06 INTERRUPT SINCE
;LAST CLEARED
;OPTION FLAGS
;DRIVE 0 TO BE TESTED FLAG
;MEMORY SIZE REPORT FLAG
;MEMORY PARITY REPORT FLAG
;START UP INSTRUCTIONS REPORTED FLAG
;PARITY PRESENT FLAG
;BSE HAS BEEN REPORTED
;FIRST PASS FORMAT SWITCH
;CP IS 11/70 FLAG
;DRIVE NUMBERS REPORTED FLAG
;LINE CLOCK PRESENT
;LIST OF DRIVES DROPPED
;WORD OF PARITY FLAGS
;POINTER TO CSR TO USE FOR SETTING BAD
;PARITY
;LINE CLOCK TICK COUNTER
;REFORMAT SWITCH FOR HALT
; THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
; HEADER OR DATA COMPARE CONTINUATION PROCESS.
;DESTINATION HOLD
;SOURCE HOLD
;WORD NUMBER IN ERROR HOLD
;WORDS LEFT IN COMPARE HOLD
;KW11-L ADDRESS
;KW11-L VECTOR
;MM11 ADDRESS
;MM11 VECTOR
;DRV TYP 0=RK06, 2000=RK07
;DRV 0 0=RK06, 2000=RK07
;DRV 1
;DRV 2
;DRV 3
;DRV 4
;DRV 5
;DRV 6
;DRV 7

```

```

2741
2742
2743
2744
2745      .SBTTL  PROGRAM SETUP
2746
2747 001742 012737 000002 001624 SETCLK: MOV    #2,SRTFLG      ;SET START FLAG FOR CLOCK ADJUST
2748 001750 000412                BR      START1
2749
2750 001752 012737 000001 001624 PARM:   MOV    #1,SRTFLG      ;SET START FLAG FOR PARMETER START
2751 001760 000406                BR      START1
2752
2753 001762 012737 177777 001624 RESTRT: MOV   #-1,SRTFLG     ;LOAD START FLAG FOR PARMETER START
2754 001770 000402                BR      START1
2755
2756 001772 005037 001624          START:  CLR    SRTFLG        ;CLEAR START FLAG
2757 001776 000005                START1: RESET                ;RESET THE WHOLE SYSTEM
2758 002000 012706 001100          MOV    #STACK,SP        ;INITIALIZE STACK POINTER
2759 002004 012746 000340          MOV    #PR7,-(SP)       ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
2760 002010 012746 002016          MOV    #1$,-(SP)       ;LOAD START OF PROGRAM
2761 002014 000002                RTI                    ;LOAD PSW
2762
2763 002016 004737 045524          1$:    JSR    PC,STKINT    ;INITIALIZE KEYBOARD
2764 002022 005037 001676          CLR    REFORMAT        ;CLEAR REFORMAT SWITCH
2765      .SBTTL  INITIALIZE THE COMMON TAGS
2766      ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2767 002026 012706 001100          MOV    #SCMTAG,R6      ;FIRST LOCATION TO BE CLEARED
2768 002032 005026                CLR    (R6)+           ;CLEAR MEMORY LOCATION
2769 002034 022706 001140          CMP    #SWR,R6        ;;DONE?
2770 002040 001374                BNE   -.6              ;;LOOP BACK IF NO
2771 002042 012706 001100          MOV    #STACK,SP      ;;SETUP THE STACK POINTER
2772      ;;INITIALIZE A FEW VECTORS
2773 002046 012737 033172 000020    MOV    #SSCOPE,@IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
2774 002054 012737 000340 000022    MOV    #340,@IOTVEC+2 ;LEVEL 7
2775 002062 012737 034212 000030    MOV    #SEERR,@EMTVEC  ;EMT VECTOR FOR ERROR ROUTINE
2776 002070 012737 000340 000032    MOV    #340,@EMTVEC+2 ;LEVEL 7
2777 002076 012737 047364 000034    MOV    #STRAP,@TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
2778 002104 012737 000340 000036    MOV    #340,@TRAPVEC+2;LEVEL 7
2779 002112 012737 047206 000024    MOV    #SPWRDN,@PWRVEC ;POWER FAILURE VECTOR
2780 002120 012737 000340 000026    MOV    #340,@PWRVEC+2 ;LEVEL 7
2781 002126 013737 032456 032450    MOV    SENDCT,SEOPCT  ;SETUP END-OF-PROGRAM COUNTER
2782 002134 005037 001262                CLR    STIMES          ;INITIALIZE NUMBER OF ITERATIONS
2783 002140 005037 001264                CLR    ESCAPE          ;CLEAR THE ESCAPE ON ERROR ADDRESS
2784 002144 112737 000001 001115    MOVB   #1,SEMAX        ;ALLOW ONE ERROR PER TEST
2785 002152 012737 002152 001106    MOV    #.,SLPADR       ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2786 002160 012737 002160 001110    MOV    #.,SLPERR       ;SETUP THE ERROR LOOP ADDRESS
2787      ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2788      ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
2789 002166 013746 000004                MOV    @ERRVEC,-(SP)   ;SAVE ERROR VECTOR
2790 002172 012737 002226 000004    MOV    #64$,@ERRVEC   ;SET UP ERROR VECTOR
2791 002200 012737 177570 001140    MOV    #DSWR,SWR       ;SETUP FOR A HARDWARE SWICH REGISTER
2792 002206 012737 177570 001142    MOV    #DDISP,DISPLAY  ;AND A HARDWARE DISPLAY REGISTER
2793 002214 022777 177777 176716    CMP    #-1,@SWR        ;TRY TO REFERENCE HARDWARE SWR
2794 002222 001012                BNE   66$              ;BRANCH IF NO TIMEOUT TRAP OCCURRED
2795      ;AND THE HARDWARE SWR IS NOT = -1
2796 002224 000403                BR     65$              ;BRANCH IF NO TIMEOUT

```

```

2797 002226 012716 002234      64$:  MOV      #65$, (SP)      ;;SET UP FOR TRAP RETURN
2798 002232 000002                RTI
2799 002234 012737 000176 001140 65$:  MOV      #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2800 002242 012737 000174 001142  MOV      #DISPREG,DISPLAY
2801 002250 012637 000004      66$:  MOV      (SP)+, @#ERRVEC  ;;RESTORE ERROR VECTOR
2802
2803 002254 005037 001304                CLR      $PASS           ;;CLEAR PASS COUNT
2804 002260 132737 000200 001317  BITB     #APTSIZE, $ENVM  ;;TEST USER SIZE UNDER APT
2805 002266 001403                BEQ      67$           ;;YES, USE NON-APT SWITCH
2806 002270 012737 001320 001140  MOV      #SSWREG, SWR    ;;NO, USE APT SWITCH REGISTER
2807 002276
2808
2809      .SBTTL  TYPE PROGRAM NAME
      ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2810 002276 005227 177777      INC      #-1           ;;FIRST TIME?
2811 002302 001066                BNE      68$           ;;BRANCH IF NO
2812 002304 022737 032612 000042  CMP      #SENDAD, @#42  ;;ACT-11?
2813 002312 001462                BEQ      68$           ;;BRANCH IF YES
2814 002314 104401 002362      TYPE     69$           ;;TYPE ASCIZ STRING
2815      .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2816 002320 005737 000042      TST      @#42           ;;ARE WE RUNNING UNDER XXDP/ACT?
2817 002324 001012                BNE      70$           ;;BRANCH IF YES
2818 002326 123727 001316 000001  CMPB    $ENV, #1        ;;ARE WE RUNNING UNDER APT?
2819 002334 001406                BEQ      70$           ;;BRANCH IF YES
2820 002336 023727 001140 000176  CMP      SWR, #SWREG    ;;SOFTWARE SWITCH REG SELECTED?
2821 002344 001005                BNE      71$           ;;BRANCH IF NO
2822 002346 104406                GTSWR
2823 002350 000403                BR       71$           ;;GET SOFT-SWR SETTINGS
2824 002352 112737 000001 001134 70$:  MOVB    #1, $AUTOB     ;;SET AUTO-MODE INDICATOR
2825 002360 71$:
2826 002360 000437      BR      68$           ;;GET OVER THE ASCIZ
2827      ;;69$:  .ASCIZ <CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MAINDEC-11-CZR6K-E*<CRLF>
2828 002460 68$:
2829 002460 132737 000200 001317  BITB    #BIT7, $ENVM   ;;TEST IF DO NOT SIZE
2830 002466 001052                BNE      3$           ;;YES - SKIP
2831 002470 004737 032664      JSR     PC, $SIZE
2832 002474 023727 033170 000740  CMP     $LSTBK, #740   ;;MAKE SURE MEMORY IS SUFFICIENT
2833 002502 103007                BHS     2$           ;;YES - SKIP
2834 002504 104401 051453      TYPE    ,OPRO05       ;;MESSAGE (NOT ENOUGH MEMORY)
2835 002510 012737 000001 032450  MOV     #1, $EOPCT     ;;FORCE END OF PROGRAM
2836 002516 000137 032422      JMP     $EOP
2837 002522
2838 002522 122737 000013 000041 2$:  CMPB    #13, @#41     ;;LOAD FROM XXDP
2839 002530 001003                BNE     99$           ;;BRANCH IF NOT
2840 002532 104401 051524      TYPE    ,OPRO07
2841 002536 000000                HALT
2842 002540
2843 002540 013700 033170      99$:  MOV     $LSTBK, R0     ;;GET LAST BANK
2844 002544 012701 000006      MOV     #6, R1        ;;SET SHIFT COUNT
2845 002550 013703 033166      MOV     $LSTAD, R3    ;;GET LAST ADDRESS
2846 002554 005004                CLR     R4            ;;CLEAR R4 FOR OVERFLOW
2847 002556 005737 032722      TST     $KT11        ;;MEM MANAGE PRESENT?
2848 002562 100005                BPL     23$           ;;NO - SKIP
2849 002564 006100      22$:  ROL     R0            ;;SHIFT BANK LEFT
2850 002566 006104                ROL     R4            ;;ADD IN CARRY
2851 002570 005301                DEC     R1            ;;DECREMENT COUNT
2852 002572 001374                BNE     22$           ;;LOOP IF NOT ZERO

```


2853	002574	050003				BIS	RO,R3	;SET BANK BITS IN LAST ADDRESS
2854	002576	112737	000001	001327	23\$:	MOVB	#1,\$MTP1	;FORCE MEMORY TYPE TO 1
2855	002604	110437	001326			MOVB	R4,\$MAMS1	;STORE UPPER MEMORY ADDRESS
2856	002610	010337	001330			MOV	R3,\$MADR1	;STORE LOWER ADDRESS
2857	002614	032737	000010	001664	3\$:	BIT	#SRTINS,OPTFLG	;TEST IF ALREADY REPORTED
2858	002622	001005				BNE	24\$;YES - SKIP
2859	002624	104401	052330			TYPE	,OPR016	;TYPE STARTUP INSTRUCTIONS
2860	002630	052737	000010	001664		BIS	#SRTINS,OPTFLG	;SET REPORTED FLAG
2861	002636				24\$:			
2862	002636	022737	000001	001624		CMP	#1,SRTFLG	;CHECK IF PARAMETER START
2863	002644	001122				BNE	15\$;NO START TESTING
2864	002646	104401	051344		5\$:	TYPE	,OPR001	;TYPE "RK611 BUS ADDRESS () ="
2865	002652	013746	001352			MOV	\$BASE,-(SP)	;SAVE \$BASE FOR TYPEOUT
2866	002656	104402				TYPOC		;GO TYPE--OCTAL ASCII(ALL DIGITS)
2867	002660	104401	051373			TYPE	,OPR002	
2868	002664	104412				RDOCT		;GET VALUE
2869	002666	012637	001222			MOV	(SP)+,\$TMPO	
2870	002672	001407				BEQ	7\$;CHECK IF <CR>
2871	002674	022737	160000	001222		CMP	#160000,\$TMPO	;CHECK IF IN I/O PAGE
2872	002702	101361				BHI	5\$	
2873	002704	013737	001222	001352		MOV	\$TMPO,\$BASE	;LOAD NEW BUS ADDRESS
2874	002712	104401	051401		7\$:	TYPE	,OPR003	;TYPE "RK611 VECTOR ADDRESS () ="
2875	002716	013746	001346			MOV	\$VECT1,-(SP)	;GET \$VECT1 FOR TYPOUT
2876	002722	042716	160000			BIC	#160000,(SP)	;CLEAR PRIORITY BITS
2877	002726	104402				TYPOC		
2878	002730	104401	051373			TYPE	,OPR002	
2879	002734	104412				RDOCT		;GET VALUE
2880	002736	012637	001222			MOV	(SP)+,\$TMPO	
2881	002742	001412				BEQ	10\$;CHECK IF <CR>
2882	002744	022737	001000	001222		CMP	#1000,\$TMPO	
2883	002752	101757				BLOS	7\$;CHECK IF LEGAL
2884	002754	042737	017777	001346		BIC	#17777,\$VECT1	;CLEAR OLD VECTOR
2885	002762	053737	001222	001346		BIS	\$TMPO,\$VECT1	;LOAD NEW VECTOR ADDRESS
2886	002770	104401	051431		10\$:	TYPE	,OPR004	;TYPE "RK611 PRIORITY () ="
2887	002774	005046				CLR	-(SP)	
2888	002776	113716	001347			MOVB	\$VECT1+1,(SP)	
2889	003002	006216				ASR	(SP)	;SHIFT 5 BITS RIGHT
2890	003004	006216				ASR	(SP)	
2891	003006	006216				ASR	(SP)	
2892	003010	006216				ASR	(SP)	
2893	003012	006216				ASR	(SP)	
2894	003014	104402				TYPOC		
2895	003016	104401	051373			TYPE	,OPR002	
2896	003022	104412				RDOCT		;GET VALUE
2897	003024	012637	001222			MOV	(SP)+,\$TMPO	
2898	003030	001430				BEQ	15\$;CHECK IF <CR>
2899	003032	022737	000007	001222		CMP	#7,\$TMPO	;CHECK IF LEGAL
2900	003040	103753				BLO	10\$	
2901	003042	022737	000004	001222		CMP	#4,\$TMPO	
2902	003050	101347				BHI	10\$	
2903	003052	006337	001222			ASL	\$TMPO	;SHIFT 5 BITS LEFT
2904	003056	006337	001222			ASL	\$TMPO	
2905	003062	006337	001222			ASL	\$TMPO	
2906	003066	006337	001222			ASL	\$TMPO	
2907	003072	006337	001222			ASL	\$TMPO	
2908	003076	042737	160000	001347		BIC	#160000,\$VECT1+1	;CLEAR OLD PRIORITY

E05

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 56
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0056

2909	003104	053737	001222	001347	BIS	\$TMP0,\$VECT1+1	;LOAD RK611 PRIORITY
2910	003112	005037	001304	15\$:	CLR	\$PASS	;SET PASS COUNT TO ZERO
2911	003116	005037	001666		CLR	DRVDRP	;CLEAR DROPPED DRIVES LIST
2912	003122	042737	004000	001664	BIC	#DRVPT,OPTFLG	;CLEAR DRIVE #'S REPORTED FLAG
2913	003130	004737	035120		JSR	PC,OPTTST	;SETUP PARITY CHECK & CLOCK
2914	003134	013700	001346		MOV	\$VECT1,RO	;STORE VECTOR FOR USE
2915	003140	042700	160000		BIC	#160000,RO	;CLEAR PRIORITY BITS
2916	003144	010037	001620		MOV	RO,RKVEC	
2917	003150	012710	035040		MOV	#INTHLR,(RO)	;SETUP INTERRUPT ADDRESS
2918	003154	113737	001347	001622	MOVB	\$VECT1+1,RKPRI	;STORE PRIORITY FOR USE
2919	003162	013702	001352		MOV	\$BASE,R2	;SET BASE ADDRESS
2920	003166	005037	001264		CLR	\$ESCAPE	;CLEAR ESCAPE
2921	003172	012746	000000		MOV	#PRO,-(SP)	;SET PRIORITY
2922	003176	012746	003204		MOV	#16\$,-(SP)	
2923	003202	000002			RTI		
2924	003204			16\$:			
2925							

F05

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 57
**BASIC INTERFACE AND OPTION TESTS

SEQ 0057

```

2926 .SBTTL **BASIC INTERFACE AND OPTION TESTS
2927 ;*****
2928 ;*TEST 1 RK611 BASE ADDRESS TEST
2929 ;* CHECK THAT READING THE RK611 BASE ADDRESS (RKCS1) DOES NOT
2930 ;* CAUSE A NON-EXISTANT MEMORY TRAP. IF A TRAP OCCURS
2931 ;* THE PROGRAM IS HALTED.
2932 ;*****
2933 ;*****
2934 †ST1: SCOPE
2935 003204 000004 MOV #100,$TIMES ;DO 100 ITERATIONS
2936 003206 012737 000100 001262 MOV #STACK,SP ;CLEAN OFF STACK
2937 003214 012706 001100 MOV #4,R1 ;SET POINTER TO VECTOR
2938 003220 012701 000004 MOV (R1)+,-(SP) ;STORE OLD VECTOR CONTENTS
2939 003224 012146 MOV (R1),-(SP)
2940 003226 011146 MOV #4,R1 ;RESET POINTER
2941 003230 012701 000004 MOV #NEXINT,(R1)+ ;SET VECTOR TO NEM TEST HANDLER
2942 003234 012721 035032 MOV #PR7,(R1) ;SET PRIORITY
2943 003240 012711 000340 MOV $BASE,R2 ;SET POINTER TO RK611 BASE ADDRESS
2944 003244 013702 001352 CLR INTSET ;CLEAR INTERRUPT COUNTER
2945 003250 005037 001662 MOV #0,RKCS1(R2) ;WRITE CS1 TO SEE IN NEM WILL SET
2946 003254 012762 000000 000000 NOP
2947 003262 000240 NOP
2948 003264 000240 NOP
2949 003266 000240 TST INTSET ;TEST IF COUNTER IS 0
2950 003270 005737 001662 BEQ IS ;YES - SKIF ERROR REPORT
2951 003274 001406 053747 001360 MOV #EM1,EMIN ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
2952 003276 012737 053747 001360 ERROR 1
2953 003304 104001 JMP CTRHLT ;GO TO CONTROLLED HALT
2954 003306 000137 044302 1$: MOV #6,R1 ;RESTORE VECTOR
2955 003312 012701 000006 MOV (SP)+,(R1)
2956 003316 012611 MOV (SP)+,-(R1)
2957
2958 ;*****
2959 ;*TEST 2 INTERRUPT VECTOR ADDRESS TEST
2960 ;* CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
2961 ;* EXPECTED ADDRESS. IF INTERRUPT VECTOR IS IN ERROR,
2962 ;* THE PROGRAM IS HALTED.
2963 ;*****
2964 ;*****
2965 †ST2: SCOPE
2966 003322 000004 MOV #100,$TIMES ;DO 100 ITERATIONS
2967 003324 012737 000100 001262 MOV #CLR,RKCS2(R2) ;CLEAR SUBSYSTEM, SPECIFICALLY TO
2968 003332 012762 005000 000010 ;CLEAR ANY OLD INTERRUPTS
2969 003340 005037 001662 CLR INTSET ;CLEAR INTERRUPT COUNTER
2970 003344 012762 000300 000000 MOV #RDY!IE,RKCS1(R2) ;WRITE CS1 TO FORCE INTERRUPT
2971 003352 000240 NOP
2972 003354 000240 NOP
2973 003356 000240 NOP
2974 003360 005737 001662 TST INTSET ;TEST IF INTERRUPT OCCURRED
2975 003364 001011 001103 BNE 3$ ;YES - SKIP ERROR REPORT
2976 003366 105737 001103 TSTB $ERFLG ;TEST IF ERFLG ALREADY SET. IF SO THE
2977 ; INTERRUPT WENT TO THE WRONG VECTOR
2978 ; AND MESSAGE HAS BEEN REPORTED.
2979 003372 001004 BNE 2$ ;THEREFORE - EXIT
2980 003374 012737 053747 001360 MOV #EM1,EMIN ;MESSAGE (NO INTERRUPT)
2981 003402 104001 ERROR 1

```

2982 003404 000137 044302
2983 003410

2\$: JMP CTRHLT ;GO TO CONTROLLED HALT
3\$:

.SBTTL **STATUS VALID TESTS

*TEST 3 SELECT ALL DRIVES

* IF NOT RUNNING IN APT AUTOMATIC ENVIRONMENT,
* DETERMINE WHAT DRIVES ARE ON-LINE BY
* SELECTING ALL DRIVES. IF NON-EXISTENT DRIVE REPORTED
* MAKE SURE STATUS VALID IS RESET. IF DRIVE
* PRESENT MAKE SURE NO ERROR EXISTS, DRIVE
* IS CYCLED UP, AND STATUS VALID SET, AND DSC RESET.

* IF RUNNING IN APT AUTOMATIC ENVIRONMENT, THE DRIVES
* IDENTIFIED IN ETABLE ARE TESTED FOR NO ERROR, DRIVE
* CYCLED UP, AND STATUS VALID SET.

* IF LOCATION 41 INDICATES THE XXDP MEDIA IS ON
* THE RK06, DRIVE 0 WILL ONLY BE TESTED IF THE PARAM
* START (214) WAS USED. IF THE AUTOMATIC START (200)
* IS USED, DRIVE 0 IS NOT TESTED. THE RESTART (204)
* WILL RETAIN THE TEST STATUS OF DRIVE 0.

* IF THE PARAM START IS USED, THE OPERATOR MUST
* EITHER PLACE DRIVE 0 OFF LINE IF IT IS NOT TO BE TESTED
* OR UNLOADED AND A SCRATCH MEDIA MOUNTED IF IT IS TO
* BE TESTED. THE PROGRAM WILL MONITOR OFF LINE AND VOLUME
* VALID TO DETERMINE THE TEST STATUS OF DRIVE 0.

* ALL DRIVES TO BE TESTED MUST BE ON-LINE, CYCLED UP, AND
* WRITE LOCK RESET. ADDRESSES OF DRIVES THAT ARE NON-
* EXISTANT EITHER BECAUSE THE DRIVE DOES NOT EXIST OR IS OFF-
* LINE ARE USED TO VERIFY NON-EXISTANT DRIVE ERROR DETECTION.
* DRIVES THAT ARE ON-LINE BUT NOT CYCLED UP OR ARE WRITE
* LOCKED ARE NOT TESTED.

* AT COMPLETION OF THE TEST A MESSAGE WILL BE GIVEN TO
* IDENTIFY THE DRIVES TO BE USED IN TESTING.

* NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE
* ANY OTHER TEST THAT FOLLOWS.
* THE DETERMINATION OF WHETHER EACH DRIVE IS AN
* RK06 OR AN RK07 IS MADE IN THIS TEST, AND A
* TABLE IS MADE ACCORDINGLY.

3029
3030 003410 000004
3031 003412 012737 000062 001262
3032 003420 005037 001720
3033 003424 104416
3034 003426 104003
3035
3036 003430 012746 000000
3037 003434 012746 003442

†ST3: SCOPE
MOV #50, \$TIMES ;:DO 50. ITERATIONS
CLR DTYPE ;:ASSUME RK06
TSSINIT ;:CALL SUBSYSTEM CLEAR AND TEST
ERROR 3
MOV #PRO, -(SP) ;:SET PROCESSOR PRIORITY TO ALLOW
MOV #1\$, -(SP) ;:RK611 INTERRUPTS

3038	003440	000002				RTI		
3039								
3040	003442	013701	001620		1\$:	MOV	RKVEC,R1	;GET VECTOR
3041	003446	012721	035040			MOV	#INTHLR,(R1)+	;LOAD INTERRUPT VECTOR
3042	003452	012711	000340			MOV	#PR7,(R1)	
3043	003456	012703	001354			MOV	#DEV0,R3	;GET ADDRESS OF DEVICE MAP
3044	003462	132737	000200	001317		BITB	#BIT7,\$ENVM	;TEST IF SHOULD SIZE
3045	003470	001007				BNE	50\$;NO - SKIP DRIVE SIZING.
3046	003472	005737	001304			TST	\$PASS	;TEST IF FIRST PASS
3047	003476	001004				BNE	50\$;NO - SKIP TO DRIVE SELECT TEST
3048	003500	032737	004000	001664		BIT	#DRVRPT,OPTFLG	;TEST IF DRIVE #'S REPORTED
3049	003506	001402				BEQ	92\$;NO - GOTO SIZING
3050	003510	000137	004164		50\$:	JMP	11\$	
3051	003514	005013			92\$:	CLR	(R3)	;CLEAR DEVICE MAP
3052	003516	123727	000041	000013		CMPB	#41,#13	;TEST IF RK06 IS LOAD DEVICE
3053	003524	001111				BNE	77\$;NO - SKIP
3054	003526	022737	000001	001624		CMP	#1,SRTFLG	;WAS START AT PARAM?
3055	003534	001411				BEQ	2\$;YES - SKIP
3056	003536	104401	051524			TYPE	,OPRO07	;NO TEST OF DRIVE 0
3057	003542	052737	000001	001666		BIS	#BIT0,DRVDRP	;SET DRIVE 0 DROPPED
3058	003550	042737	000001	001664		BIC	#DOTST,OPTFLG	;DR FLAG - NO TEST DRIVE 0
3059	003556	000477				BR	7\$	
3060	003560	104401	051524		2\$:	TYPE	,OPRO06	;MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3061	003564	005037	001610			CLR	,L.CS2	;SET TO DRIVE 0
3062	003570	005037	001232			CLR	\$TMP4	;CLEAR FOR USE AS A SWITCH
3063	003574	012737	000101	001600	3\$:	MOV	#SELDRV,L.CS1	;LOAD FOR SELECT
3064	003602	012737	003670	001264		MOV	#4\$, \$ESCAPE	;SET UP ESCAPE IN CASE OF ERR
3065	003610	104417				TLOADRK		;LOAD RK & DO SELECT
3066								
3067	003612	104423				TWAT16		;WAIT 16MS FOR COMPLETION
3068	003614	104002				ERROR	2	;NOT DONE ON TIME
3069								
3070	003616	104420				TGETRK		;GET RK REGISTER
3071	003620	032737	100000	001540		BIT	#CERR,T.CS1	;TEST IF CERR
3072	003626	001431				BEQ	5\$;NO - SKIP
3073	003630	032737	010000	001550		BIT	#NED,T.CS2	;TEST IF NEC
3074	003636	001014				BNE	4\$;YES - SKIP
3075	003640	032737	000040	001554		BIT	#DTYE,T.ER	;ELSE SEE IF DRIVE TYPE ERROR
3076	003646	001406				BEQ	26\$;BR IF NO
3077								
3078	003650	012737	002000	001720		MOV	#CDT,DTYPE	;ELSE SETUP FOR RK07
3079								
3080	003656	104416				TSSINIT		;INIT SUBSYS
3081	003660	104003				ERROR	3	;NOT SUCCESSFUL
3082	003662	000744				BR	3\$;AND TRY AGAIN
3083								
3084	003664	104421			26\$:	TCHKOP		;CHECK THE OPERATION AND REPORT THE ERROR
3085	003666	104004				ERROR	4 ;OR5,6,7	;AFTER THE ERROR IS REPORTED THE TEST
3086								;IS ABORTED
3087	003670	104401	051524		4\$:	TYPE	,OPRO07	;TYPE NO TEST OF DRIVE 0
3088	003674	042737	000001	001664		BIC	#DOTST,OPTFLG	;DR FLAG - NO TEST OF DRIVE 0
3089	003702	052737	000001	001666		BIS	#BIT0,DRVDRP	;SET DRV 0 AS DROPPED
3090	003710	000422				BR	7\$;SKIP OVER WAIT FOR PACK MOUNT
3091	003712	005737	001232		5\$:	TST	\$TMP4	;TEST FLAG DRIVE READY HAS RESET
3092	003716	001010				BNE	6\$;YES - SKIP TO CHECK IF IT IS SET AGAIN
3093	003720	032737	000200	001552		BIT	#DRDY,T.DS	;ELSE CHECK READY

```

3094 003726 001322          BNE      3$          ;STILL SET - GET STATUS AGAIN
3095 003730 012737 177777 001232  MOV      #-1,$TMP4  ;ELSE SET FLAG TO INDICATE READY WENT LOW
3096 003736 000716          BR       3$          ;GO GET STATUS AGAIN
3097
3098 003740 032737 000200 001552 6$:  BIT      #DRDY,T.DS ;TEST IF READY SET AGAIN
3099 003746 001712          BEQ      3$          ;NO - GO GET STATUS AGAIN
3100 003750 052737 000001 001664 77$: BIS      #DOTST,OPTFLG ;ELSE SET DRV 0 TEST FLAG.
3101
3102 003756 012737 004064 001264 7$:  MOV      #35$,$ESCAPE ;SET UP ESCAPE IN CASE OF ERR
3103 003764 005000          CLR      RO         ;CLEAR FOR DRIVE NUMBER COUNTER
3104 003766 012701 000001          MOV      #1,R1      ;SET BIT 0 AS DRIVE SELECTOR
3105 003772 005037 001720          CLR      DTYPE     ;INIT TO RK06
3106
3107 003776 032737 000001 001664          BIT      #DOTST,OPTFLG ;TEST DRIVE 0?
3108 004004 001440          BEQ      9$          ;NO - SKIP
3109
3110 004006 104416          8$:  TSSINIT ;INITIALIZE SUBSYSTEM
3111 004010 104003          ERROR  3          ;ERROR IF NOT SUCCESSFUL
3112
3113 004012 010037 001610          MOV      RO,L.CS2  ;LOAD DRIVE NUMBER
3114 004016 012737 000101 001600  MOV      #SELDRV,L.CS1 ;LOAD DRIVE SELECT
3115 004024 104417          TLOADRK ;LOAD RK REGS
3116
3117 004026 104423          TWAT16 ;WAIT FOR INTERRUPT
3118 004030 104002          ERROR  2          ;TO SLOW/NOT COMPLETE ERROR
3119
3120 004032 104420          TGETRK ;GET RK REGS
3121 004034 032737 100000 001540  BIT      #CERR,T.CS1 ;ERROR?
3122 004042 001031          BNE     10$         ;YES - SKIP
3123 004044 032737 000200 001552  BIT      #DRDY,T.DS ;ELSE TEST IF DRIVE READY
3124 004052 001404          BEQ     35$         ;NO - SKIP
3125 004054 032737 004000 001552  BIT      #WRL,T.DS  ;ELSE TEST IF WRITE LOCKED
3126 004062 001403          BEQ     36$         ;NO - SKIP
3127 004064 050137 001666          35$: BIS      R1,DRVDRP ;SET THIS BIT AS DROPPED DRIVE
3128 004070 000406          BR      9$
3129 004072          36$:
3130 004072 050113          BIS     R1,(R3)    ;SET BIT - DRIVE PRESET IN MAP
3131 004074 006300          ASL     RO
3132 004076 013760 001720 001722  MOV      DTYPE,TYPTBL(RO) ;SET TABLE WITH DRIVE TYPE INFO
3133 004104 006200          ASR     RO
3134
3135 004106 005200          9$:  INC     RO         ;BUMPS TO NEXT DRIVE
3136 004110 006301          ASL     R1        ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3137 004112 032701 000400          BIT      #BIT8,R1  ;WAS LAST DRIVE DONE?
3138 004116 001022          BNE     11$         ;BR IF YES
3139 004120 005037 001720          CLR      DTYPE    ;ELSE INIT FOR RK06 TO TEST NEXT DRV
3140 004124 000730          BR      8$
3141
3142 004126 032737 010000 001550 10$: BIT      #NED,T.CS2 ;WAS CERR DUE TO NED?
3143 004134 001364          BNE     9$         ;YES - BUMP TO NEXT DRIVE
3144 004136 032737 000040 001554  BIT      #DTYPE,T.ER ;ELSE SEE IF DRIVE TYPE ERR
3145 004144 001404          BEQ     24$        ;BR IF NO
3146
3147 004146 012737 002000 001720  MOV      #CDT,DTYPE ;ELSE SET UP FOR RK07
3148 004154 000714          BR      8$         ;AND TRY AGAIN
3149

```

```

3150 004156 104421          24$:  TCHKOP          ;ELSE REPORT THE ERRORS
3151 004160 104004          ERROR      4 ;OR5,6,7
3152 004162 000000          .WORD     0
3153 004164 032737 004000 001664 101$:  BIT      #DRV RPT,OPTFLG ;TEST IF SHOULD REPORT
3154 004172 001037          BNE      16$          ;NO - SKIP
3155
3156 004174 005713          TST      (R3)         ;ANY DRIVE AVAILABLE?
3157 004176 001004          BNE      12$         ;BR IF NOT ZERO
3158 004200 104401 051605          TYPE     OPR008      ;ELSE REPORT NO DRIVES AVAILABLE
3159 004204 000137 044302          JMP      CTRHLT      ;GO TO CONTROLLED HALT
3160
3161 004210 012701 000200          12$:  MOV      #BIT7,R1   ;SET DRIVE SELECTOR FOR DRIVE 7
3162 004214 012700 000007          MOV      #7,R0       ;SET DRIVE NUMBER TO 7
3163 004220 104401 051670          TYPE     ,OPR009     ;TYPE PREFIX TO DRIVE TEST MESSAGE
3164
3165 004224 030113          13$:  BIT      R1,(R3)   ;TEST IF THIS DRIVE TO BE TESTED
3166 004226 001007          BNE      15$         ;YES - SKIP
3167
3168 004230 005300          14$:  DEC      R0       ;ELSE DECREMENT DRIVE NUMBER
3169 004232 006201          ASR     R1          ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3170 004234 001373          BNE     13$        ;IF NOT SHIFTED OUT - LOOP
3171 004236 052737 004000 001664          BIS     #DRV RPT,OPTFLG ;SET DRIVE #'S REPORTED FLAG
3172 004244 000412          BR      16$         ;ELSE GO TO STATUS VALID TEST
3173
3174 004246 010037 004162          15$:  MOV      R0,101$    ;PUT DRIVE NUMBER IN TYPE LOCATION
3175 004252 052737 000060 004162          BIS     #BIT4:BITS,101$ ;MAKE IT ASCIZ
3176 004260 104401          TYPE     TYPE IT
3177 004262 004162          101$
3178 004264 104401 051341          TYPE     SPACE2     ;TYPE SOME SPACES
3179 004270 000757          BR      14$         ;LOOP
3180
3181 004272 005000          16$:  CLR      R0       ;CLEAR DRIVE NUMBER COUNTER
3182 004274 012701 000001          MOV     #1,R1       ;SET DRIVE SELECTOR TO DRIVE 0
3183 004300 012737 177777 001240          MOV     #-1,$TMP7    ;SET $TMP7 NEGATIVE
3184 004306 012737 177777 001630          MOV     #-1,DRVBIT   ;SET DRIVE SELECT BIT NEGATIVE
3185 004314 012737 004422 001264          MOV     #18,$ESCAPE  ;SET ESCAPE IN CASE OF ERR
3186 004322 030137 001666          BIT     R1,DRVDRP   ;HAS DRIVE 0 BEEN DROPPED?
3187 004326 001035          BNE     18$         ;YES - SKIP TO DRIVE INC.
3188
3189 004330 104416          17$:  TSSINIT          ;CLEAR SUBSYSTEM
3190 004332 104003          ERROR   3          ;ERROR FOR BAD CLEAR
3191
3192 004334 010037 001610          MOV     R0,L.CS2    ;SET DRIVE SELECT
3193 004340 010037 001626          MOV     R0,DRVNUM   ;SET DRIVE NUMBER
3194 004344 004737 036512          JSR     PC,GTYP0    ;GET DRV TYPE
3195 004350 012737 000101 001600          MOV     #SELDRV,L.CS1 ;SET FOR DRIVE SELECT
3196
3197 004356 104417          TLOADRK          ;LOAD RK REGS
3198 004360 104423          TWAT16          ;WAIT FOR INTERRUPT
3199 004362 104002          ERROR   2          ;ERROR TO SLOW/NOT COMPLETE
3200 004364 030113          BIT     R1,(R3)     ;WAS THAT DRIVE AVAILABLE
3201 004366 001026          BNE     19$         ;YES - SKIP
3202
3203 004370 104422          TCHKWE          ;CHECK THAT ERROR OCCURRED
3204 004372 000000          .WORD   0          ;NONE OF GROUP 1
3205 004374 000000          .WORD   0          ;NONE OF GROUP 2

```



```

3262 004624 023727 001624 000002      CMP      SRTFLG,#2      ;TEST IF CLOCK ADJUST START
3263 004632 001002                BNE      25$           ;NO - SKIP
3264 004634 000137 044172          JMP      ADJCLK        ;GO TO ADJUST CLOCK ROUTINE
3265
3266 004640                25$:
3267
3268      ;*****
3269      ;*TEST 4      RELEASE ALL DRIVES
3270      ;*
3271      ;*      RELEASE ALL DRIVES. MAKE SURE NO ERROR
3272      ;*      SETS AND STATUS VALID IS RESET.
3273      ;*
3274      ;*****
3275 004640 000004      TST4:  SCOPE
3276 004642 012737 000062 001262      MOV      #50.,$TIMES  ;:DO 50. ITERATIONS
3277 004650 104416                TSSINIT                ;:INITIALIZE SUBSYSTEM
3278 004652 104003                ERROR  3              ;:BAD INIT
3279
3280 004654 013737 001626 001610      MOV      DRVNUM,L.CS2 ;:SET DRIVE NUMBER
3281 004662 012737 000101 001600      MOV      #SELDLV,L.CS1 ;:SET DRIVE SELECT
3282
3283 004670 104417                TLOADRK                ;:LOAD RK REGS
3284 004672 104423                TWAT16                 ;:WAIT FOR INTERRUPT
3285 004674 104002                ERROR  2              ;:TO SLOW/NOT COMPLETE ERROR
3286
3287 004676 104421                TCHKOP                 ;:CHECK FOR ANY ERRORS
3288 004700 104004                ERROR  4 ;OR5,6,7    ;:REPORT ANY ERRORS
3289
3290 004702 012737 000010 001610      MOV      #RLS,L.CS2   ;:SET DRIVE RELEASE,STILL SET FOR SELECT
3291
3292 004710 104417                TLOADRK                ;:LOAD RK REGS
3293 004712 104423                TWAT16                 ;:WAIT FOR INTERRUPT
3294 004714 104002                ERROR  2              ;:TO SLOW/NOT COMPLETE ERROR
3295
3296 004716 104421                TCHKOP                 ;:CHECK FOR ANY ERRORS
3297 004720 104004                ERROR  4 ;OR 5, 6, OR 7 ;:REPORT ALL ERRORS
3298 004722 032737 100000 001552      BIT      #SVAL,T.DS   ;:DID SVAL RESET?
3299 004730 001404                BEQ      1$           ;:YES - SKIP
3300 004732 012737 056231 001400      MOV      #EM49,EM3N   ;:MESSAGE (SVAL NOT RESET W/RELEASE)
3301 004740 104003                ERROR  3
3302
3303      1$:
3304 004742 004737 036412      TSTLUP: JSR      PC,LDCON ;:LOAD RK06-RK07 CONSTANTS
3305      ;*****
3306      ;*TEST 5      NON-STANDARD MESSAGES AND SVAL
3307      ;*
3308      ;*      PICK ONE OF THE AVAILABLE DRIVES AND GET
3309      ;*      NON-STANDARD MESSAGES. MAKE SURE NO
3310      ;*      ERROR OCCURS AND STATUS VALID DOES NOT SET
3311      ;*      AND THAT NON-STANDARD MESSAGES CAUSE STATUS
3312      ;*      VALID TO RESET.
3313      ;*
3314      ;*****
3315 004746 000004      TST5:  SCOPE
3316 004750 012737 000062 001262      MOV      #50.,$TIMES  ;:DO 50. ITERATIONS
3317 004756 104416                TSSINIT                ;:CLEAR SUBSYSTEM

```

M05

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 64
TS NON-STANDARD MESSAGES AND SVAL

SEQ 0064

```

3318 004760 104003          ERROR 3          ;BAD CLEAR MESSAGE
3319 004762 012701 000001  MOV      #1,R1      ;PRESET R1 FOR MESSAGE PAIR 1
3320 004766 013737 001626 001610  MOV      DRVNUM,L.CS2 ;LOAD DRV NUMBER
3321 004774 012737 000101 001600  MOV      #SELD, L.CS1 ;LOAD SELECT COMMAND
3322 005002 005037 001616          CLR      L.MR1      ;LOAD FOR STANDARD STATUS
3323 005006 104417          TLOADRK ;LOAD RK
3324 005010 104423          TWAT16  ;WAIT FOR INTERRUPT
3325 005012 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
3326 005014 104421          TCHKOP  ;CHECK OPERATION
3327 005016 104004          ERROR 4 ;5,6 OR 7 ;REPORT ALL ERRORS
3328
3329 005020 032737 100000 001552  BIT      #SVAL,T.DS  ;TEST STATUS VALID SET
3330 005026 001007          BNE     2$         ;YES-SKIP
3331
3332 005030 012737 057006 001460  MOV      #EMSV,EM11N
3333 005036 012737 050536 061454  MOV      #EMSEL,DF011A
3334 005044 104011          ERROR 11         ;"SVAL NOT SET RESULT OF DRIVE SELECT"
3335
3336 005046 010137 001616          2$: MOV      R1,L.MR1   ;LOAD MESSAGE PAIR SELECT
3337
3338 005052 104417          TLOADRK ;LOAD RK
3339 005054 104423          TWAT16  ;WAIT FOR INTERRUPT
3340 005056 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
3341
3342 005060 104421          TCHKOP  ;CHECK OPERATION
3343 005062 104004          ERROR 4 ;5,6, OR 7 ;REPORT ALL ERRORS
3344
3345 005064 032737 100000 001552  BIT      #SVAL,T.DS  ;TEST STATUS VALID RESET
3346 005072 001407          BEQ     3$         ;YES-SKIP
3347
3348 005074 012737 057006 001470  MOV      #EMSV,EM12N
3349 005102 012737 057023 061454  MOV      #EMNZPR,DF011A
3350 005110 104012          ERROR 12         ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"
3351
3352 005112 022701 000003          3$: CMP      #3,R1     ;WAS PAIR 3 SELECTED?
3353 005116 001402          BEQ     4$         ;YES-SKIP
3354 005120 005201          INC     R1         ;BUMP TO NEXT PAIR
3355 005122 000727          BR     1$         ;SKIP TO DO IT.
3356 005124
3357
3358 *****
3359 *TEST 6 WRITING CS2 AND STATUS VALID
3360 *
3361 * SELECT AN AVAILABLE DRIVE. MAKE SURE STATUS
3362 * VALID IS SET. WRITE COMMAND AND STATUS REGISTER 2.
3363 * MAKE SURE STATUS VALID RESETS.
3364 *****
3365 005124 000004          †ST6: SCOPE
3366 005126 012737 000062 001262  MOV      #50,$TIMES ;DO 50. ITERATIONS
3367 005134 104416          TSSINIT ;CLEAR SUBSYSTEM
3368 005136 104003          ERROR 3          ;BAD INIT ERROR
3369
3370 005140 013737 001626 001610  MOV      DRVNUM,L.CS2 ;LOAD DRIVE NUMBER
3371 005146 012737 000101 001600  MOV      #SELD, L.CS1 ;LOAD DRIVE SELECT
3372
3373 005154 104417          TLOADRK ;LOAD RK

```

N05

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 65
T6 WRITING CS2 AND STATUS VALID

SEQ 0065

```

3374 005156 104423      TWAT16      ;WAIT FOR INTERRUPT
3375 005160 104002      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3376                                TCHKOP      ;CHECK OPERATION
3377 005162 104421      ERROR      4 ;5,6, OR 7 ;REPORT ALL ERRORS
3378 005164 104004
3379
3380 005166 032737 100000 001552      BIT          #SVAL,T.DS      ;TEST STATUS VALID SET
3381 005174 001007      BNE        1$      ;YES-SKIP
3382
3383 005176 012737 057006 001460      MOV          #EMSVAL,EM11N
3384 005204 012737 050536 061454      MOV          #EMSELD,DF011A
3385 005212 104011      ERROR      11      ;"SVAL NOT SET RESULT OF DRV SELECT"
3386
3387 005214 013762 001626 000010 1$:      MOV          DRVNUM,RKCS2(R2) ;WRITE CS2 TO RESET SVAL
3388
3389 005222 104420      TGETRK      ;GET RK REGS.
3390
3391 005224 032737 100000 001552      BIT          #SVAL,T.DS      ;TEST SVAL RESET
3392 005232 001407      BEQ        2$      ;YES-SKIP
3393
3394 005234 012737 057006 001470      MOV          #EMSVAL,EM12N
3395 005242 012737 057064 061454      MOV          #EMWCS2,DF011A
3396 005250 104012      ERROR      12      ;"SVAL NOT RESET BY WRITING CS2"
3397 005252
3398
    
```

CZ
CZ

3399
3400
3401
3402
3403
3404
3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454

005252 000004
005254 012737 000062 001262
005262 104416
005264 104003
005266 013737 001626 001610
005274 012737 000101 001600
005302 053737 036506 001600
005310 104417
005312 104423
005314 104002
005316 104422
005320 000040
005322 000000
005324 000000
005326 104004
005330 032737 100000 001552
005336 001007
005340 012737 057006 001460
005346 012737 055502 061454
005354 104011
005356
005356 000004
005360 012737 000062 001262
005366 104416

```
.SBTTL **CONTROLLER ERROR TESTS
*****
*TEST 7 DRIVE TYPE ERROR
*
* CREATE A DRIVE TYPE ERROR. MAKE SURE DRIVE
* TYPE ERROR SETS AND STATUS VALID SETS.
*****
↑ST7: SCOPE
MOV #50.,$TIMES ;;DO 50. ITERATIONS
TSSINIT ;;CLEAR SUBSYSTEM
ERROR 3 ;;BAD INIT ERROR
MOV DRVNUM,L.CS2 ;;LOAD DRIVE NUMBER
MOV #SELDRV,L.CS1 ;;LOAD DRIVE SELECT
BIS HOLD2,L.CS1 ;;LOAD DRIVE TYPE
TLOADRK ;;LOAD RK
TWAT16 ;;WAIT FOR INTERRUPT
ERROR 2 ;;TO SLOW/NOT COMPLETE ERROR
TCHKWE ;;CHECK OPERATION WITH EXPECTED ERROR
.WORD 000040 ;;DRIVE TYPE ERROR
.WORD 0
.WORD 0
ERROR 4 ; OR 5,6,7 ;;REPORT ANY DIFFERENCES (NO ERRORS,
;;ADDITIONAL ERRORS, DIFFERENT ERRORS)
BIT #SVAL, T.DS ;;TEST IF SVAL SET
BNE 1$ ;;YES-SKIP
MOV #EMSVAL,EM11N
MOV #EMDTPE,DF011A
ERROR 11 ;"SVAL NOT SET RESULT OF DRV TYPE ERR"
1$:
*****
*TEST 10 STATUS VALID AND PARITY ERROR
*
* ISSUE A SELECT TO AN AVAILIABLE DRIVE WITH BAD PARITY.
* MAKE SURE SPAR, CONTROLLER ERROR, ATTENTION,
* DRIVE STATUS CHANGES, DRPAR, DRIVE INTERRUPT,
* AND STATUS VALID SET. ISSUE A CONTROLLER
* CLEAR. MAKE SURE DRIVE INTERRUPT AND ATTENTION
* ARE STILL SET. SELECT DRIVE AGAIN WITH GOOD
* PARITY. MAKE SURE ATTENTION, DRIVE STATUS
* CHANGE, DRPAR, CONTROLLER ERROR, DRIVE INTERRUPT,
* AND STATUS VALID ARE SET AND SPAR IS RESET.
* ISSUE A CONTROLLER CLEAR. GET NON-STANDARD MESSAGES
* AND MAKE SURE ONLY DRIVE INTERRUPT AND ATTENTION
* ARE SET. CLEAR ATTENTION WITH DRIVE CLEAR. REPEAT
* FOR ALL AVAILIABLE DRIVES.
*****
↑ST10: SCOPE
MOV #50.,$TIMES ;;DO 50. ITERATIONS
TSSINIT ;;CLEAR SUBSYSTEM
```

3455	005370	104003				ERROR 3		;BAD INIT ERROR
3456								
3457	005372	013737	001626	001610		MOV DRVNUM,L.CS2		;LOAD DRIVE NUMBER
3458	005400	012737	000101	001600		MOV #SELDLV,L.CS1		;LOAD DRIVE SELECT
3459	005406	012737	000020	001616		MOV #PAT,L.MR1		;LOAD EVEN PARITY BIT
3460								
3461	005414	104417				TLOADRK		;LOAD RK REGS-SELECT W/EVEN PARITY
3462	005416	104423				TWAT16		;WAIT FOR INTERRUPT
3463	005420	104002				ERROR 2		;TO SLOW/NOT COMPLETE ERROR
3464								
3465	005422	104422				TCHKWE		;CHECK OPERATION FOR EXPECTED ERROR
3466	005424	000011				DRPARERR:SPARERR		;DRIVE SELECTED DRIVE BUS PARITY ERROR
3467	005426	000000				.WORD 0		;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
3468	005430	000000				.WORD 0		
3469	005432	104004				ERROR 4 ; OR 5,6,7		;REPORT ANY DIFFERENCES
3470								
3471	005434	012700	000400			MOV #BIT8,RO		;ROUTINE TO DETERMINE WHICH BIT
3472								
3473	005440	013701	001626			MOV DRVNUM,R1		;SHOULD BE SET IN ASOF TO INDICATE
3474	005444	001403				BEQ 3\$;DRIVE ATTENTION. RO WILL HAVE THE
3475	005446	006300			2\$:	ASL RO		;BIT THAT SHOULD BE SET FOR THE DRIVE
3476	005450	005301				DEC R1		;IN USE
3477	005452	001375				BNE 2\$		
3478								
3479	005454	030037	001556		3\$:	BIT RO,T.ASOF		;TEST ATTENTION SET
3480	005460	001007				BNE 4\$;YES-SKIP
3481	005462	012737	057172	001460		MOV #EMDA,EM11N		
3482	005470	012737	055417	061454		MOV #EMDPAR,DF011A		
3483	005476	104011				ERROR 11		;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3484	005500	032737	040000	001540	4\$:	BIT #DI,T.CS1		;TEST DRIVE INTERRUPT SET
3485	005506	001007				BNE 5\$;YES-SKIP
3486	005510	012737	057126	001460		MOV #EMDI,EM11N		
3487	005516	012737	055417	061454		MOV #EMDPAR,DF011A		
3488	005524	104011				ERROR 11		;"DRV INT NOT SET RESULT OF DRV PARITY ERR"
3489								
3490	005526	032737	040000	001552	5\$:	BIT #DSC,T.DS		;TEST DRIVE STATUS CHANGE SET
3491	005534	001007				BNE 6\$;YES-SKIP
3492	005536	012737	057146	001460		MOV #EMDSC,EM11N		
3493	005544	012737	055417	061454		MOV #EMDPAR,DF011A		
3494	005552	104011				ERROR 11		;"DSC NOT SET RESULT OF DRV PARITY ERR"
3495								
3496	005554	032737	100000	001552	6\$:	BIT #SVAL,T.DS		;TEST STATUS VALID SET
3497	005562	001007				BNE 7\$;YES-SKIP
3498	005564	012737	057006	001460		MOV #EMSVL,EM11N		
3499	005572	012737	055417	061454		MOV #EMDPAR,DF011A		
3500								
3501	005600	104011				ERROR 11		;"SVAL NOT SET RESULT OF DRV PAR ERR"
3502								
3503	005602	005037	001616		7\$:	CLR L.MR1		;CLEAR PAT IN MR1
3504								
3505	005606	052737	100000	001600		BIS #CCLR,L.CS1		;CLEAR CONTROLLER
3506	005614	104417				TLOADRK		;LOAD RK REGS TO DO CLEAR
3507								
3508								
3509	005616	104421				TCHKOP		;CHECK NO ERRORS SET
3510	005620	104004				ERROR 4 ; OR 5,6,7		;REPORT ALL ERRORS STILL SET

```

3511          005622 030037 001556          BIT      RO,T.ASOF      ;TEST ATTENTION STILL SET
3512          005626 001007          BNE      8$           ;YES-SKIP
3513          005630 012737 057172 001510      MOV      #EMDA,EM14N
3514          005636 012737 057212 061454      MOV      #EMCCLR,DF011A
3515          005644 104014          ERROR    14          ;"DRV ATT RESET RESULT OF CONT CLR"
3516          005646 032737 040000 001540 8$:      BIT      #DI,T.CS1      ;TEST DRIVE INTERRUPT STILL SET
3517          005654 001007          BNE      9$           ;YES-SKIP
3518          005656 012737 057126 001510      MOV      #EMDI,EM14N
3519          005664 012737 057212 061454      MOV      #EMCCLR,DF011A
3520          005672 104014          ERROR    14          ;"DRV INT RESET RESULT OF CONTROLLER CLR"
3521          005674 012737 000101 001600 9$:      MOV      #SELDLV,L.CS1 ;LOAD SELECT DRIVE
3522          005702 104417          TLOADRK          ;LOAD RK REGS
3523          005704 104423          TWAT16          ;WAIT FOR INTERRUPT
3524          005706 104002          ERROR    2          ;TO SLOW/NOT COMPLETE ERROR
3525          005710 104422          TCHKWE          ;CHECK THAT ERRORS ARE SET
3526          005712 000010          .WORD    000010      ;DPAR SET (NO SPAR 1, REPORT
3527          005714 000000          .WORD    0           ;ANY DISCREPENCIES
3528          005716 000000          .WORD    0           ;
3529          005720 104004          ERROR    4 ; OR 5,6,7
3530          005722 030037 001556          BIT      RO,T.ASOF      ;TEST ATTENTION STILL SET.
3531          005726 001007          BNE      10$          ;YES-SKIP
3532          005730 012737 057172 001460      MOV      #EMDA,EM11N
3533          005736 012737 055417 061454      MOV      #EMDPAR,DF011A
3534          005744 104011          ERROR    11          ;"DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3535          005746 032737 040000 001540 10$:     BIT      #DI,T.CS1      ;TEST DRIVE INTERRUPT STILL SET
3536          005754 001007          BNE      11$          ;YES-SKIP
3537          005756 012737 057126 001460      MOV      #EMDI,EM11N
3538          005764 012737 055417 061454      MOV      #EMDPAR,DF011A
3539          005772 104011          ERROR    11          ;"DRV INT NOT SET RESULT OF DRV PARITY ERR"
3540          005774 032737 040000 001552 11$:     BIT      #DSC,T.DS      ;TEST STATUS CHANGE STILL SET
3541          006002 001007          BNE      12$          ;YES-SKIP
3542          006004 012737 057146 001460      MOV      #EMDSC,EM11N
3543          006012 012737 055417 061454      MOV      #EMDPAR,DF011A
3544          006020 104011          ERROR    11          ;"DSC NOT SET RESULT OF DRV PARITY ERR"
3545          006022 032737 100000 001552 12$:     BIT      #SVAL,T.DS      ;TEST STATUS VALID STILL SET
3546          006030 001007          BNE      13$          ;YES-SKIP
3547          006032 012737 057006 001460      MOV      #EMSVAL,EM11N
3548          006040 012737 055417 061454      MOV      #EMDPAR,DF011A
3549          006046 104011          ERROR    11          ;"SVAL NOT SET RESULT OF DRV PARITY ERR"
3550          006050 052737 100000 001600 13$:     BIS      #CCLR,L.CS1    ;CLEAR CONTROLLER
3551          006056 104417          TLOADRK          ;LOAD RK REGS (DO CLEAR)
3552          006060 012701 000001          MOV      #1,R1         ;SET COUNTER TO SELECT STATUS PAIR
3553          006064 012737 000101 001600      MOV      #SELDLV,L.CS1 ;LOAD DRIVE SELECT
3554
3555
3556

```

E06

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 69
T10 STATUS VALID AND PARITY ERROR

SEQ 0069

```

3567 006072 010137 001616      14$:  MOV      R1,L.MR1      ;LOAD STATUS PAIR SELECTION
3568 006076 104417                TLOADRK                ;LOAD RK REGS
3569 006100 104423                TWAT16                 ;WAIT FOR INTERRUPT
3570 006102 104002                ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
3571
3572 006104 104421                TCHKOP                ;CHECK IF ANY ERRORS SET
3573 006106 104004                ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS SET.
3574
3575 006110 030037 001556                BIT      RO,T.ASOF     ;TEST ATTENTION STILL SET
3576 006114 001007                BNE      15$           ;YES-SKIP
3577 006116 012737 057172 001410                MOV      #EMDA,EM4N
3578 006124 012737 057023 061454                MOV      #EMNZPR,DF011A
3579 006132 104014                ERROR      14          ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
3580
3581 006134 032737 040000 001540      15$:  BIT      #DI,T.CS1
3582 006142 001007                BNE      16$
3583 006144 012737 057126 001510                MOV      #EMDI,EM14N
3584 006152 012737 057023 061454                MOV      #EMNZPR,DF011A
3585 006160 104014                ERROR      14          ;"DRV INT RESET RESULT OF NON-0 PAIR SELECT"
3586
3587 006162 005201                16$:  INC      R1           ;BUMP PAIR SELECT
3588 006164 022701 000004                CMP      #4,R1        ;ALL PAIRS DONE?
3589 006170 001340                BNE      14$          ;NO-LOOP
3590
3591 006172 005037 001616                CLR      L.MR1        ;CLEAR MR1
3592
3593 006176 012737 000105 001600                MOV      #CLEAR,L.CS1 ;LOAD DRIVE CLEAR
3594
3595 006204 104417                TLOADRK                ;DO DRIVE CLEAR
3596 006206 104423                TWAT16                 ;WAIT FOR INTERRUPT
3597 006210 104002                ERROR      2          ;TO SLOW/NOT COMPLETE ERROR
3598
3599 006212 104421                TCHKOP                ;CHECK FOR ANY ERRORS
3600 006214 104004                ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS
3601
3602 006216 012701 000020                17$:  MOV      #20,R1      ;SET COUNT FOR SHORT WAIT
3603 006222 005301                DEC      R1           ;TO ALLOW CONTROLLER TIME TO POLL
3604 006224 001376                BNE      17$         ;DRIVES
3605
3606 006226 104420                TGETRK                ;GET RK REGS
3607 006230 030037 001556                BIT      RO,T.ASOF     ;TEST ATTENTION RESET
3608 006234 001407                BEQ      18$          ;YES-SKIP
3609 006236 012737 057172 001470                MOV      #EMDA,EM12N
3610 006244 012737 050564 061454                MOV      #EMDCLR,DF011A
3611 006252 104012                ERROR      12          ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
3612
3613 006254 032737 040000 001540      18$:  BIT      #DI,T.CS1
3614 006262 001407                BEQ      19$
3615 006264 012737 057126 001470                MOV      #EMDI,EM12N
3616 006272 012737 050564 061454                MOV      #EMDCLR,DF011A
3617 006300 104012                ERROR      12          ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
3618
3619 006302                19$:
3620                ;*****
3621                ;*TEST 11      UNIT FIELD ERROR ON RELEASE
3622                ;*

```

F06

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 70
T11 UNIT FIELD ERROR ON RELEASE

SEQ 0070

```

3623 : *      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
3624 : *      DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
3625 : *      A RELEASE COMMAND.  CLOCK THROUGH PHASE ADDRESS 2.
3626 : *      TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
3627 : *      ERROR SETS.
3628 : *
3629 : *****
3630 006302 000004      †ST11: SCOPE
3631 006304 012737 000062 001262  MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3632 006312 104416      TSSINIT      ;;CLEAR SUBSYSTEM
3633 006314 104002      ERROR      2      ;;BAD INIT ERROR
3634
3635 006316 013737 001626 001610  MOV      DRVNUM,L.CS2    ;;SELECT A DRIVE
3636 006324 012737 000101 001600  MOV      #SELDRV,L.CS1  ;;DO DRIVE SELECT
3637
3638 006332 104417      TLOADRK      ;;LOAD RK
3639 006334 104423      TWAT16      ;;WAIT FOR INTERRUPT
3640 006336 104002      ERROR      2      ;;TO SLOW/NOT COMPLETE ERROR
3641
3642 006340 104421      TCHKOP      ;;CHECK FOR ANY ERRORS
3643 006342 104004      ERROR      4 ; OR 5,6,7 ;;REPORT ALL ERRORS.
3644
3645 006344 052737 000010 001610  BIS      #RLS,L.CS2     ;;LOAD RELEASE
3646 006352 012737 000040 001616  MOV      #DMD,L.MR1     ;;SET DIAGNOSTIC MODE
3647
3648 006360 104417      TLOADRK      ;;LOAD RK
3649
3650 006362 004437 036674      JSR      R4,MCLOCK      ;;CALL MAINT CLOCK
3651 006366      023      .BYTE      †D19      ;;NUMBER OF PHASES
3652 006367      002      .BYTE      2      ;;NUMBER OF CLOCK XISTIONS
3653
3654 006370 042762 000040 000026  BIC      #DMD,RKMR1(R2) ;;CLEAR DIAG MODE
3655
3656 006376 104423      TWAT16      ;;WAIT FOR INTERRUPT
3657 006400 104002      ERROR      2      ;;TO SLOW/NOT COMPLETED
3658
3659 006402 104422      TCHKWE      ;;CHECK OPERATION WITH ERROR
3660 006404 000000      .WORD      0
3661 006406 000000      .WORD      0
3662 006410 000004      .WORD      UFERR      ;;UNIT FIELD ERROR
3663 006412 104004      ERROR      4 ; OR 5,6,7 ;;REPORT ANY DISCREPENCIES
3664
3665 006414 104416      TSSINIT      ;;CLEAR SUBSYSTEM TO INSURE UFE RESETS
3666 006416 104002      ERROR      2
3667 : *****
3668 : †TEST 12      UNIT FIELD ERROR ON SELECT
3669 : *
3670 : *      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
3671 : *      DRIVE.  PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE
3672 : *      A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE
3673 : *      SELECTED = 0.  CLOCK THROUGH PHASE ADDRESS 6.
3674 : *      TURN OFF DIAGNOSTIC MODE.  MAKE SURE UNIT FIELD
3675 : *      ERROR SETS.
3676 : *
3677 : *****
3678 006420 000004      †ST12: SCOPE

```



```

3679 006422 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3680 006430 104416                      TSSINIT                      ;CLEAR SUBSYSTEM
3681 006432 104003                      ERROR    3                  ;BAD INIT ERROR
3682
3683 006434 013737 001626 001610      MOV      DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3684 006442 012737 000101 001600      MOV      #SELDV,L.CS1      ;LOAD DRIVE SELECT
3685
3686 006450 104417                      TLOADRK                      ;LOAD RK
3687 006452 104423                      TWAT16                       ;WAIT FOR INTERRUPT
3688 006454 104002                      ERROR    2                  ;TO SLOW/NOT COMPLETE
3689
3690 006456 104421                      TCHKOP                       ;CHECK FOR ANY ERROR
3691 006460 104004                      ERROR    4 ; OR 5,6,7      ;REPORT ALL ERRORS
3692
3693 006462 012737 000043 001616      MOV      #DMD!BIT1!BIT0,L.MR1 ;LOAD DIAG MODE & MSG PAIR 3
3694 006470 005037 001610                      CLR      L.CS2              ;LOAD FOR DRIVE 0
3695
3696 006474 104417                      TLOADRK                      ;LOAD RK
3697
3698 006476 004437 036674                      JSR      R4,MCLOCK          ;CALL MAINTENANCE CLOCK
3699 006502 026                          .BYTE 1D22                  ;THROUGH PHASE 6
3700 006503 002                          .BYTE 2                      ;PLUS 2 TRANSITIONS
3701
3702 006504 042762 000040 000026      BIC      #DMD,RKMR1(R2)    ;CLEAR DIAG MODE
3703
3704 006512 104423                      TWAT16                       ;WAIT FOR INTERRUPT
3705 006514 104002                      ERROR    2                  ;TO SLOW/NOT COMPLETED ERROR
3706
3707 006516 013737 036510 006526      MOV      HOLD3,1$          ;EXPECTED ERROR
3708
3709 006524 104422                      TCHKWE                       ;CHECK OPERATION WITH ERROR
3710 006526 000000                      1$: .WORD 0
3711 006530 000000                      .WORD 0
3712 006532 000004                      .WORD UFERR                  ;UNIT FIELD ERROR SHOULD SET
3713 006534 104004                      ERROR    4 ; OR 5,6,7      ;REPORT ANY DISCREPENCIES
3714
3715 .SBTTL **ATTENTION HANDLING BY CONTROLLER
3716
3717 ;*****
3718 ;*TEST 13      DOUBLE INTERRUPT
3719 ;*
3720 ;*      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECALIBRATE.
3721 ;*      MAKE SURE STATUS VALID IS SET.  CHECK THAT SECOND
3722 ;*      INTERRUPT OCCURS.  AFTER SECOND INTERRUPT
3723 ;*      CHECK THAT STATUS VALID IS RESET.  ISSUE SELECT
3724 ;*      AND MAKE SURE STATUS VALID SETS.  CLEAR DRIVE.
3725 ;*      CHECK THAT DRIVE STATUS CHANGE SETS
3726 ;*      (BIT 14 OF DRIVE STATUS
3727 ;*      REGISTER)
3728 ;*
3729 ;*****
3730 006536 000004                      ST13: SCOPE
3731 006540 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3732 006546 104416                      TSSINIT                      ;CLEAR SUBSYSTEM
3733 006550 104003                      ERROR    3                  ;BAD INIT ERROR
3734

```

```

3735 006552 013737 001626 001610      MOV      DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3736 006560 012737 000113 001600      MOV      #RECAL,L.CS1     ;LOAD RECAL
3737
3738 006566 104417      TLOADRK      ;LOAD RK
3739 006570 104423      TWAT16      ;WAIT FOR 1ST INTERRUPT
3740 006572 104002      ERROR      2      ;TO SLOW/NOT COMPLETE
3741 006574 005037 001662      CLR      INTSET      ;CLEAR INTERRUPT FLAG
3742 006600 104420      TGETRK      ;GET RK REGS
3743 006602 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST SVAL SET
3744 006610 001010      BNE      1$      ;YES-SKIP
3745 006612 012737 057006 001460      MOV      #EMSVAL,EM11N
3746 006620 012737 050625 061454      MOV      #EMRCAL,DF011A
3747 006626 104011      ERROR      11      ;"SVAL NOT SET RESULT OF RECAL"
3748 006630 000463      BR        50$      ;ABORT TEST
3749
3750 006632 104437      1$:      TWAT8S      ;WAIT FOR INTERRUPT
3751 006634 000401      BR        2$      ;NO INTERRUPT RETURN
3752 006636 000404      BR        3$      ;INTERRUPT RETURN
3753
3754 006640 012737 056301 001370 2$:      MOV      #EM50,EM2N      ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
3755 006646 104002      ERROR      2
3756
3757 006650 104420      3$:      TGETRK      ;GET RK REGS
3758 006652 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST SVAC SET NOW
3759 006660 001410      BEQ      4$      ;NO-SKIP
3760 006662 012737 057006 001470      MOV      #EMSVAL,EM12N
3761 006670 012737 057233 061454      MOV      #EM2INT,DF011A
3762 006676 104012      ERROR      12      ;"SVAL NOT RESET RESULT OF SECOND TEST"
3763 006700 000437      BR        50$
3764
3765 006702 032737 040000 001552 4$:      BIT      #DSC,T.DS      ;TEST DSC SET BY ATTENTION
3766 006710 001010      BNE      5$      ;YES-SKIP
3767 006712 012737 057146 001460      MOV      #EMDSC,EM11N
3768 006720 012737 057233 061454      MOV      #EM2INT,DF011A
3769 006726 104011      ERROR      11      ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
3770 006730 000423      BR        50$
3771
3772 006732 012737 000101 001600 5$:      MOV      #SELDRV,L.CS1   ;LOAD DRIVE SELECT
3773
3774 006740 104417      TLOADRK      ;LOAD RK REGS
3775 006742 104423      TWAT16      ;WAIT FOR INTERRUPT
3776 006744 104002      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3777
3778 006746 104421      TCHKOP      ;CHECK FOR ANY ERRORS
3779
3780 006750 104004      ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS
3781
3782 006752 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST SVAC SET
3783 006760 001007      BNE      50$      ;YES-SKIP
3784 006762 012737 057006 001460      MOV      #EMSVAL,EM11N
3785 006770 012737 050536 061454      MOV      #EMSELD,DF011A
3786 006776 104011      ERROR      11      ;"SVAL NOT SET RESULT OF DRV SEL.
3787 007000      50$:
3788      ;*****
3789      ;*TEST 14      SINGLE INTERRUPT FROM ATTENTION
3790      ;*

```

```

3791      ;*      DO A SEEK TO CYLINDER 0.  WAIT FOR INTERRUPT FROM
3792      ;*      DRIVE ATTENTION.  LOWER PRIORITY AGAIN AND MAKE
3793      ;*      SURE ANOTHER INTERRUPT DOES NOT OCCUR.  CLEAR DRIVE.
3794      ;*
3795      ;*****
3796 007000 000004      TST14: SCOPE
3797 007002 012737 000062 001262  MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
3798 007010 104416      TSSINIT      ;;CLEAR SUBSYSTEM
3799 007012 104003      ERROR      3      ;;BAD INIT ERROR
3800
3801 007014 013737 001626 001610  MOV      DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3802 007022 012737 000117 001600  MOV      #SEEK,L.CS1      ;LOAD SEEK DCYL LEFT AT 0.
3803
3804 007030 104417      TLOADRK      ;LOAD RK REGS
3805 007032 104423      TWAT16      ;WAIT FOR INTERRUPT
3806 007034 104002      ERROR      2      ;TO SLOW/NOT COMPLETED ERROR
3807
3808
3809 007036 104420      TGETRK      ;GET RK REGS
3810
3811 007040 032737 040000 001540  BIT      #DI,T.CS1      ;TEST DI SET
3812 007046 001010      BNE      2$      ;YES-SKIP
3813 007050 012737 057126 001460  MOV      #EMDI,EM11N
3814 007056 012737 057254 061454  MOV      #EMSKSF,DF011A
3815 007064 104011      ERROR      11      ;"DI NOT SET RESULT OF SEEK TO SELF"
3816 007066 000417      BR      50$
3817
3818
3819 007070 012700 000031      2$:  MOV      #25.,RO      ;LOAD AND DECREMENT A COUNT TO
3820 007074 005300      3$:  DEC      RO      ;ZERO. GIVE CONTROLLER A CHANCE TO
3821 007076 001376      BNE      3$      ;INTERRUPT AGAIN.  ERROR IF IT DOES.
3822
3823 007100 022737 000001 001662  CMP      #1,INTSET      ;CHECK ONLY ONE INTERRUPT OCCURRED
3824 007106 001407      BEQ      50$      ;YES-SKIP
3825 007110 012737 057426 001500  MOV      #EMMI,EM13N
3826 007116 012737 057254 061454  MOV      #EMSKSF,DF011A
3827 007124 104013      ERROR      13      ;"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"
3828
3829 007126 104421      50$: TCHKOP      ;CHECK FOR ANY ERRORS
3830 007130 104004      ERROR      4 ;OR 5,6,7 ;REPORT ALL ERRORS
3831      ;*****
3832      ;*TEST 15      RESET ATTENTIONS WITH UNIBUS INIT
3833      ;*
3834      ;*      DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
3835      ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
3836      ;*
3837      ;*****
3838 007132 000004      TST15: SCOPE
3839 007134 012737 000005 001262  MOV      #5.,$TIMES      ;;DO 5. ITERATIONS
3840 007142 104416      TSSINIT      ;;CLEAR SUBSYSTEM
3841 007144 104003      ERROR      3      ;;BAD INIT ERROR
3842
3843 007146 013737 001626 001610  MOV      DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3844 007154 012737 000117 001600  MOV      #SEEK,L.CS1      ;LOAD SEEK (TO SELF-0)
3845
3846 007162 104417      TLOADRK      ;LOAD RK REGS

```

```

3847 007164 104423          TWAT16          ;WAIT FOR INTERRUPT
3848 007166 104002          ERROR 2          ;TO SLOW/NOT COMPLETE
3849
3850 007170 104420          TGETRK          ;GET RK REGS
3851
3852 007172 032737 040000 001540  BIT #DI,T.CS1    ;TEST DI SET
3853 007200 001010          BNE 1$          ;YES-EXIT
3854 007202 012737 057126 001460  MOV #EMDI,EM11N
3855 007210 012737 057254 061454  MOV #EMSKSF,DF011A
3856 007216 104011          ERROR 11        ;"DI NOT SET RESULT OF SEEK TO SELF
3857 007220 000450          BR 50$
3858
3859 007222 005037 001662          1$: CLR INTSET      ;CLEAR INTERRUPT COUNTER
3860 007226 000005          RESET          ;DO UNIBUS RESET
3861 007230 004737 045524          JSR PC,$TKINT  ;RESET KEYBOARD INTERRUPT
3862
3863 007234 005037 001674          CLR LCLKTK     ;CLEAR TICK COUNTER
3864 007240 004737 035672          JSR PC,MYTIME  ;CALL TIMER
3865 007244 022737 000012 001674  CMP #10.,LCLKTK ;COUNT 10 TICKS (MILLISECONDS)?
3866 007252 001372          BNE 5$          ;NO - LOOP
3867
3868 007254 012762 000100 000000  MOV #IE,RKCS1(R2) ;SET IE FOR ANY STRAY INTERRUPTS
3869 007262 004737 035120          JSR PC,OPTTST  ;SET UP OPTIONS AGAIN
3870
3871 007266 104423          TWAT16          ;WAIT 16 MS FOR AN INTERRUPT
3872 007270 000410          BR 2$          ;NONE IS EXPECTED SO RETURN SHOULD BE
3873                                     ;HERE-BR TO CONTINUE TEST.
3874 007272 012737 057271 001500  MOV #EMUXIT,EM13N ;INT OCCURRED ON RESET
3875 007300 012737 057333 061454  MOV #EMRSET,DF011A
3876 007306 104013          ERROR 13        ;"UNEXECUTED INTERRUPT RESULT OF RESET"
3877 007310 000414          BR 50$
3878 007312 104420          TGETRK          ;GET RK REGS
3879 007314 032737 040000 001540  BIT #DI,T.CS1    ;TEST DI RESET
3880 007322 001407          BEQ 50$         ;YES-SKIP
3881 007324 012737 057126 001470  MOV #EMDI,EM12N
3882 007332 012737 057333 061454  MOV #EMRSET,DF011A
3883 007340 104012          ERROR 12        ;"DI NOT RESET RESULT OF RESET"
3884
3885 007342          50$:
3886
3887 .SBTTL **ILLEGAL DISK ADDRESS ERROR TESTS
3888
3889 ;*****
3890 ;*TEST 16 ILLEGAL DISK ADDRESS (PART 1)
3891 ;*
3892 ;* ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
3893 ;* ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
3894 ;* CLEAR CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7,
3895 ;* CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
3896 ;* HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
3897 ;*
3898 ;*****
3899 TST16: SCOPE
3900 MOV #10,$TIMES ;DO 10. ITERATIONS
3901 MOV #3,R1 ;PRESET FOR SELECTING TRACK 3
3902

```

```

3903 007356 104416 TSSINIT ;CLEAR SUBSYSTEM
3904 007360 104003 ERROR 3
3905
3906 007362 012737 000113 001600 MOV #RECAL,L.CS1 ;SET UP TO RECAL
3907 007370 013737 001626 001610 MOV DRVNUM,L.CS2 ;LOAD DRIVE
3908
3909 007376 104417 TLOADRK ;LOAD RK REGS
3910
3911 007400 104423 TWAT16 ;WAIT FOR 1ST INTERRUPT
3912 007402 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3913
3914 007404 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
3915
3916 007410 104437 TWAT8S ;WAIT FOR INTERRUPT
3917 007412 104002 ERROR 2
3918
3919 007414 012737 007422 001110 MOV #1$,SLPERR ;SET LOCAL LOOP ON ERROR
3920
3921 007422 104416 1$: TSSINIT ;CLEAR SUBSYSTEM
3922 007424 104003 ERROR 3 ;BAD INIT ERROR
3923
3924 007426 013737 001626 001610 MOV DRVNUM,L.CS2 ;LOAD DRIVE NUMBER
3925 007434 012737 000117 001600 MOV #SEEK,L.CS1 ;LOAD SEEK
3926 007442 110137 001607 MOV#B R1,L.DT ;LOAD TRACK
3927
3928 007446 104417 TLOADRK ;LOAD RK REGS
3929 007450 104423 TWAT16 ;WAIT FOR INTERRUPT
3930 007452 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
3931
3932 007454 032701 000001 BIT #BIT0,R1 ;TEST IF HEAD ADDRESS HAS BIT 0
3933 007460 001403 BEQ 2$ ;NO - SKIP
3934 007462 032701 000002 BIT #BIT1,R1 ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3935 007466 001007 BNE 3$ ;YES-GO CHECK BOTH IDAE AND SKI SET
3936
3937 007470 104422 2$: TCHKWE ;CHECK OPERATION WITH ERROR
3938 007472 002000 IDAERR ;ILLEGAL DISK ADDRESS ERROR
3939 007474 000000 0
3940 007476 000000 0
3941 007500 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL DISCREPANCIES
3942 007502 104415 SCOP1 ;LOCAL LOOP ON ERROR
3943 007504 000406 BR 4$
3944
3945 007506 104422 3$: TCHKWE ;CHECK OPERATION WITH ERROR
3946 007510 002002 IDAERR!SKIERR ;ILLEGAL DISK ADDRESS ERROR
3947 007512 000000 0 ;SEEK INCOMPLETE
3948 007514 000000 0
3949 007516 104004 ERROR 4 ;OR 5,6,7 ;REPORT ANY DISCREPANCIES
3950 007520 104415 SCOP1 ;LOCAL LOOP ON ERROR TO 1$
3951
3952 007522 005201 4$: INC R1 ;ELSE BUMP TO NEXT ILLEGAL TRACK
3953 007524 022701 000010 CMP #8.,R1 ;ALL ILLEGAL TRACKS SELECTED?
3954 007530 001334 BNE 1$ ;NO-LOOP
3955
3956 ;*****
3957 ;*TEST 17 ILLEGAL DISK ADDRESS (PART 2)
3958 ;*
;* ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE

```

L06

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 76
T17 ILLEGAL DISK ADDRESS (PART 2)

SEQ 0076

```

3959
3960
3961
3962
3963
3964 007532 000004
3965 007534 012737 000012 001262
3966 007542 104416
3967 007544 104003
3968
3969 007546 012737 000113 001600
3970 007554 013737 001626 001610
3971
3972 007562 104417
3973
3974 007564 104423
3975 007566 104002
3976
3977 007570 005037 001662
3978
3979 007574 104437
3980 007576 104002
3981
3982 007600 012737 007606 001110
3983
3984 007606 104416 1$:
3985 007610 104003      TSSINIT
                        ERROR 3
3986
3987 007612 013737 001626 001610
3988 007620 012737 000117 001600
3989 007626 005737 001720
3990 007632 001402
3991
3992
3993
3994 007634 000137 007666
3995
3996
3997
3998
3999 007640 012737 001000 001614 2$:
4000 ;      MOV      #1000,L.DCYL      ;LOAD DK06 ILL CYL
                        MOV      #IDAERR,4$      ;EXP RK06 ERR
4001
4002 007646 104417 3$:
4003 007650 104423      TLOADRK      ;LOAD RK REGS
4004 007652 104002      TWAT16      ;WAIT FOR INTERRUPT
                        ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
4005
4006 007654 104422 4$:
4007 007656 002000      TCHKWE      ;CHECK OPERATION WITH ERROR
                        .WORD IDAERR      ;DISK ADDRESS ERROR
4008 007660 000000      .WORD 0
4009 007662 000000      .WORD 0
4010 007664 104004      ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPANCIES
4011
4012 .SBTTL **WRITE HEADER TESTS
4013
4014
;*****

```

ILLEGAL DISK ADDRESS ERROR SETS. CLEAR CONTROLLER AND DRIVE
THIS TEST IS BYPASSED FOR THE RK07 BECAUSE THE RK07
CONTROLLER DOES NOT RECOGNIZE THE ILLEGAL DISK ADDRESS

TST17: SCOPE ;DO 10. ITERATIONS
MOV #10.,\$TIMES ;CLEAR SUBSYSTEM
TSSINIT ;BAD INIT ERROR
ERROR 3

MOV #RECAL,L.CS1 ;LOAD RECALIBRATE
MOV DRVNUM,L.CS2 ;LOAD DRIVE

TLOADRK ;LOAD RK REGS

TWAT16 ;WAIT FOR 1ST INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

CLR INTSET ;CLEAR INTERRUPT FLAG

TWAT8S ;WAIT FOR INTERRUPT
ERROR 2

MOV #1\$,SLPERR ;SET LOOP TO BYPASS RECAL

1\$: TSSINIT ;CLEAR SUBSYSTEM
ERROR 3

MOV DRVNUM, L.CS2 ;LOAD DRIVE NUMBER

MOV #SEEK,L.CS1 ;LOAD SEEK

TST DTYPE ;SEE IF RK06

BEQ 2\$;BR IF YES

MOV #1777,L.DCYL ;ELSE LOAD RK07 ILL CYL

MOV #<IDAERR!SKIERR>,4\$;EXP RK07 ERR

BR 3\$

JMP TST20 ;RK07 CONTROLLER

DOES NOT RECOGNIZE

ILLEGAL DISK ADDRESS

REV 006 THE ABOVE THREE LINES

2\$: MOV #1000,L.DCYL ;LOAD DK06 ILL CYL

MOV #IDAERR,4\$;EXP RK06 ERR

3\$: TLOADRK ;LOAD RK REGS

TWAT16 ;WAIT FOR INTERRUPT

ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

4\$: TCHKWE ;CHECK OPERATION WITH ERROR

.WORD IDAERR ;DISK ADDRESS ERROR

.WORD 0

.WORD 0

ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPANCIES

.SBTTL **WRITE HEADER TESTS

```

4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038 007666 000004
4039 007670 012737 000001 001262
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057 007676 105037 007744
4058 007702 005000
4059 007704 005005
4060 007706 013703 001640
4061 007712 012737 007720 001110
4062 007720 104416
4063 007722 104003
4064
4065 007724 013737 036476 007746
4066
4067 007732 004437 036252
4068 007736 000121
4069 007740 177400
4070 007742 070414

```

```

*TEST 20 READ BAD SECTOR INFORMATION
*
* ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
* TRACK 2 TO GET THE FACTORY DETECTED BAD
* SECTOR FILE, 26 SECTOR MODE.
*
* CYLINDER 1456 IS USED FOR THE RK07.
*
* IF AN ERROR OCCURS, READ SECTOR 2, 4, 6, OR 10(8) UNTIL
* A SUCCESSFUL READ IS DONE. IF NONE READ SUCCESSFULLY
* REMOVE THIS DRIVE FROM TEST. WHEN A READ IS SUCCESSFUL,
* TEST THAT THE PACK IS NOT AN ALIGNMENT PADK AND
* STORE THE ENTRIES FOR LATER USE.
*
* REPEAT THIS SERIES OF OPERATIONS FOR FACTORY DETECTED
* BAD SECTORS 24 SECTOR MODE, SOFTWARE DETECTED
* BAD SECTORS 26 SECTOR MODE, AND SOFTWARE DETECTED BAD
* SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR
* 24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED
* FROM TESTING.
*
*****
*ST20: SCOPE
MOV #1,$TIMES ;;DO 1 ITERATION
TSSINIT ;CLEAR SUBSYS FROM LAST IDAERR
ERROR 3 ;BAD INIT ERROR
MOV #RECAL,L.CS1 ;DO RECAL CMD
MOV DRVNUM,L.CS2
TLOADRK ;LOAD "L" REGS INTO RK REGS
TWAT16
ERROR 2 ;NO INTR
CLR INTSET ;CLR INIT FLAG
TWAT8S ;WAIT FOR 2'N INTR
ERROR 2 ;NOT THERE
CLRB 2$ ;CLEAR SECTOR POINTER
CLR R0
CLR R5 ;CLEAR R5 FOR BAD SECTOR COUNTING
MOV BSF26P,R3 ;SET POINT IN TO STORE BS 26 SECT FORMAT
MOV #1$,$LPERR ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
1$: TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
MOV LSTCYL,15$
JSR R4,LRLOAD ;LOAD "L" REGS WITH
RDATA ; READ DATA
-400 ; WORD COUNT
IBUFF ; BUFFER ADDRESS

```

NO6

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 78
T20 READ BAD SECTOR INFORMATION

SEQ 0078

4071	007744	000		2\$:	.BYTE	0		:	SECTOR ADDRESS
4072	007745	002			.BYTE	2		:	TRACK ADDRESS
4073	007746	000000		15\$:	0			:	CYLINDER ADDRESS 632/1456
4074									
4075	007750	104417			TLOADRK			:	LOAD "L" REGS INTO RK
4076	007752	104431			TWAT112			:	WAIT FOR INTERRUPT
4077	007754	104002			ERROR	2		:	TO SLOW/NOT COMPLETE ERROR
4078	007756	104421			TCHKOP			:	CHECK FOR ANY ERRORS
4079	007760	104004			ERROR	4	; OR 5,6,7	:	REPORT ALL ERRORS
4080	007762	104415			SCOPI			:	LOOP TO 1\$ IF SW 9 SET
4081									
4082	007764	105737	001103		TSTB	\$ERFLG		:	TEST FOR ERROR IN OPERATION
4083	007770	001502			BEQ	7\$:	NO-SKIP
4084	007772	005700			TST	RO		:	GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
4085	007774	001023			BNE	3\$:	NO-SKIP
4086	007776	062737	000002	007744	ADD	#2,2\$:	NEXT SECTOR ADDRESS
4087	010004	122737	000012	007744	CMPB	#10.,2\$:	PAST APPLICABLE SECTORS?
4088	010012	001066			BNE	6\$:	NO-SKIP
4089	010014	012737	056400	001360	MOV	#EM51,EMIN		:	
4090	010022	104001			ERROR	1		:	"CANNOT READ BS FILES"
4091	010024	043737	001630	001354	25\$:	BIC	DRVBIT,\$DEVM	:	CLEAR DRIVE FROM DRIVE MAP
4092	010032	053737	001630	001666		BIS	DRVBIT,DRVDRP	:	SET DRIVE DROPPED
4093	010040	000137	030346			JMP	NEWDRV	:	ABORT TEST PASS.
4094									
4095	010044	022700	000001		3\$:	CMP	#1,RO	:	GETTING A BS SOFT SECTOR 26 SECT FORMAT?
4096	010050	001014				BNE	4\$:	NO-SKIP
4097	010052	062737	000002	007744		ADD	#2,2\$:	NEXT SECTOR ADDRESS
4098	010060	122737	000026	007744		CMPB	#22.,2\$:	PAST APPLICABLE SECTORS?
4099	010066	001040				BNE	6\$:	NO-SKIP
4100	010070	012737	056400	001360		MOV	#EM51,EMIN	:	
4101	010076	104001				ERROR	1	:	"CANNOT READ BS FILES"
4102	010100	000751				BR	25\$		
4103									
4104	010102	022700	000002		4\$:	CMP	#2,RO	:	GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
4105	010106	001014				BNE	5\$:	NO-SKIP
4106	010110	062737	000002	007744		ADD	#2,2\$:	NEXT SECTOR ADDRESS
4107	010116	122737	000013	007744		CMPB	#11.,2\$:	PAST APPLICABLE SECTORS?
4108	010124	001021				BNE	6\$:	NO-SKIP
4109									
4110	010126	012737	056400	001360		MOV	#EM51,EMIN	:	
4111	010134	104001				ERROR	1	:	"CANNOT READ BS FILES"
4112	010136	000732				BR	25\$		
4113									
4114	010140	062737	000002	007744	5\$:	ADD	#2,2\$:	NEXT SECTOR (BS SOFT 24 SECT MODE)
4115	010146	122737	000027	007744		CMPB	#23.,2\$:	PAST APPLICABLE SECTORS?
4116	010154	001005				BNE	6\$:	NO-SKIP
4117	010156	012737	056400	001360		MOV	#EM51,EMIN	:	
4118	010164	104001				ERROR	1	:	"CANNOT READ BS FILES"
4119	010166	000716				BR	25\$		
4120									
4121	010170	105037	001103		6\$:	CLRB	\$ERFLG	:	CLEAR ERROR FLAG
4122	010174	000651				BR	1\$:	DO NEXT READ
4123									
4124	010176	005737	070422		7\$:	TST	IBUFF+6	:	CHECK FOR ALIGNMENT PACK
4125	010202	001405				BEQ	8\$:	NO-SKIP
4126	010204	012737	056473	001360		MOV	#EM52,EMIN	:	


```

4127 010212 104001      ERROR 1      ;"ALIGNMENT PACK. DRIVE ABORTING"
4128 010214 000703      BR      25$
4129
4130 010216 012701 070424  8$:  MOV      #IBUFF+10,R1      ;SET TO START OF BAD SECTOR DATA
4131
4132 010222 022711 177777  9$:  CMP      #-1,(R1)          ;TEST IF WORD ALL ONES (END OF DATA)
4133 010226 001422      BEQ      11$              ;YES-SKIP
4134 010230 012123      MOV      (R1)+,(R3)+      ;STORE CYLINDER
4135 010232 012123      MOV      (R1)+,(R3)+      ;TRACK AND SECTOR
4136 010234 005205      INC      R5               ;BUMP ERROR COUNTER
4137 010236 022705 000025  CMP      #21.,R5          ;DOES IT TOTAL 20 FOR THIS FORMAT?
4138 010242 001367      BNE      9$              ;NO-TEST AND MORE NEXT ADDRESS
4139 010244 012737 056571 001360  MOV      #EMS3,EMIN
4140 010252 104001      ERROR 1
4141 010254 043737 001630 001354 10$:  BIC      DRVBIT,$DEVMD      ;TO MANY BAD SECTORS
4142 010262 053737 001630 001666  BIC      DRVBIT,DRVDRP      ;CLEAR DRIVE FROM TESTING
4143 010270 000137 030346      JMP      NEWDRV            ;SET DRIVE DROPPED
4144                                ;ABORT PASS
4145 010274 005200      11$:  INC      R0               ;BUMP TO NEXT
4146 010276 022700 000001  CMP      #1 R0            ;NOW TESTING BS SOFT 26 SECTOR MODE?
4147 010302 001011      BNE      12$              ;NO-SKIP
4148 010304 012723 177777  MOV      #-1,(R3)+        ;INSERT END OF FIELD FLAG
4149 010310 010337 001644  MOV      R3,BSS26P        ;SET POINTER TO BAD SECTOR SOFTWARE FIELD
4150 010314 112737 000012 007744  MOV      #12,2$           ;SET TO FIRST SECTOR THIS MODE
4151 010322 000137 007720  JMP      1$                ;GO READ IT
4152 010326 022700 000002  12$:  CMP      #2 R0            ;NOW TESTING BS FACT 24 SECTOR MODE?
4153 010332 001014      BNE      13$              ;NO-SKIP
4154 010334 012723 177777  MOV      #-1,(R3)+        ;INSERT END OF FIELD FLAG
4155 010340 112737 000001 007744  MOV      #1,2$           ;SET TO FIRST SECTOR THIS MODE
4156 010346 010537 001646  MOV      R5,BS26CT        ;STORE TOTAL BS COUNT 26 SECTOR MODE
4157 010352 005005      CLR      R5               ;CLEAR COUNTER FOR COUNTING 24 SECT BS
4158 010354 013703 001636  MOV      BSF24P,R3        ;SET POINTER FOR STORING BS
4159 010360 000137 007720  JMP      1$                ;GO READ
4160
4161 010364 022700 000003  13$:  CMP      #3 R0            ;NOW TESTING BS SOFT 24 SECTOR MODE
4162 010370 001011      BNE      14$              ;NO-SKIP
4163 010372 012723 177777  MOV      #-1,(R3)+        ;INSERT END OF FIELD FLAG
4164 010376 010337 001642  MOV      R3,BSS24P        ;STORE POINTER TO BSS 24 SECTOR MODE
4165 010402 112737 000013 007744  MOV      #13,2$           ;GET START OF FIELDS BSS 24 SECT MODE
4166 010410 000137 007720  JMP      1$                ;GO READ IT
4167
4168 010414 012723 177777  14$:  MOV      #-1,(R3)+        ;INSERT END OF FIELD FLAG
4169 010420 010537 001650  MOV      R5,BS24CT        ;STORE COUNT BSS 24 SECTOR MODE
4170
4171 010424 012700 061506  MOV      #BS26,R0          ;GET START OF BAD SECTOR BUFFER
4172                                MOV      #BS24+52.,R3      ;GET END OF BUFFER
4173
4174
4175                                27$:  CMP      #312,(R0)+       ;TEST IF ANY SECTORS BAD IN CYL 312
4176                                BEQ      26$              ;YES - GET OUT
4177
4178                                CMP      R0,R3            ;CHECK IF ALL OF BUFFER TESTED
4179                                BNE      27$              ;NO - LOOP
4180                                BR      28$              ;EXIT
4181                                CLR      R3               ;COUNT
4182 010430 005003      27$:  CMP      #-1,(R0)         ;AT THE END OF TABLE ?
4182 010432 022710 177777

```

```

4183 010436 001404          BEQ      16$          ;BRANCH IF SO
4184 010440 022720 000312    CMP      #312,(R0)+    ;CYL 312 BAD ?
4185 010444 001372          BNE      27$          ;BRANCH IF NOT TRY AGAIN
4186 010446 000424          BR       26$          ;ELSE ERROR
4187 010450 005203          16$:    INC      R3          ;
4188 010452 020327 000001    CMP      R3,#1        ;SEE IF 1 ST HALF OF BS 26 TABLE
4189 010456 001003          BNE      17$          ;BRANCH IF NOT
4190 010460 013700 001644    MOV      BSS26P,R0    ;MOVE UP THE TABLE
4191 010464 000762          BR       27$          ;
4192 010466 020327 000002    17$:    CMP      R3,#2        ;SEE IF DID 2 ND HALF OF BS26 TABLE
4193 010472 001003          BNE      18$          ;BRANCH IF NOT
4194 010474 012700 061632    MOV      #BS24,R0    ;ELSE MOVE UP 1 ST HALF
4195 010500 000754          BR       27$          ;
4196 010502 020327 000003    18$:    CMP      R3,#3        ;SEE IF DID 1 ST HALF OF BS24 TABLE
4197 010506 001003          BNE      19$          ;BRANCH IF NOT
4198 010510 013700 001642    MOV      BSS24P,R0  ;ELSE MOVE UP 2ND HALF OF BS24
4199 010514 000746          BR       27$          ;
4200 010516 000406          19$:    BR       28$          ;ALL DONE
4201
4202 010520 012737 057452 001360 26$:    MOV      #DRVABT,EMIN ;"BAD SECTOR IN AREA FOR TEST"
4203 010526 104001          ERROR   1
4204 010530 000137 010024    JMP      25$
4205
4206 010534          28$:
4207
4208          ;*****
4209          ;*TEST 21          FORMAT IN 26 SECTOR FORMAT
4210          ;*
4211          ;*          FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
4212          ;*          FORMAT.  VERIFY FORMAT AND THAT DATA LATE DID NOT
4213          ;*          OCCUR WITH WRITE HEADER OR IN READING DATA
4214          ;*          BUFFER AFTER READ HEADER.
4215          ;*
4216          ;*****
4217 010534 000004          †ST21:  SCOPE
4218 010536 012737 000012 001262    MOV      #10, $TIMES  ;:DO 10. ITERATIONS
4219 010544 012737 000312 001676    MOV      #312,REFMT  ;:SET REFORMAT SWITCH
4220 010552 105037 010675          CLR      10$         ;:CLEAR SECTOR POINTER
4221 010556 105037 010601          CLR      11$         ;:CLEAR TRACK POINTER
4222 010562 104416          TSSINIT              ;:CLEAR SUBSYSTEM
4223 010564 104003          ERROR   3           ;:BAD INIT ERROR
4224
4225 010566 004437 036252    9$:    JSR      R4,LRLOAD   ;:LOAD "L" REGS
4226 010572 000127          WRHEAD              ;:WRITE HEADER
4227 010574 177676          -102                ;:WORD CNT FOR 26 SECTOR MODE
4228 010576 072414          OBUFF               ;:BUFFER
4229 010600 000          .BYTE   0           ;:SECTOR 0
4230 010601 000          .BYTE   0           ;:TRACK 0
4231 010602 000312          11$:    312              ;:CYL 0
4232
4233 010604 004437 042636          JSR      R4,GENCOM   ;:GENERATE DATA
4234 010610 000600          000600              ;:BUILD HEADERS-NO BAD SECTORS
4235 010612 012737 010622 001110    MOV      #111$, $LPERR ;:SET LOCAL LOOP ON ERROR
4236 010620 000402          BR       1$         ;:SKIP INIT
4237 010622
4238 010622 104416          111$:    TSSINIT          ;:CLEAR SUBSYSTEM

```

4239	010624	104003				ERROR	3					;BAD INIT ERROR
4240	010626	104417			1\$:	TLOADRK						;LOAD RK REGS
4241	010630	104431				TWAT112						;WAIT FOR INTERRUPT
4242	010632	104002				ERROR	2					;TO SLOW/NOT COMPLETE ERROR
4243												
4244	010634	104421				TCHKOP						;CHECK FOR ANY ERRORS
4245	010636	104004				ERROR	4 ; OR 5,6,7					;REPORT ALL ERRORS
4246												
4247	010640	104415				SCOPI						;INTERNAL LOOP TO 111\$
4248												
4249	010642	012737	010702	001110		MOV	#112\$,SLPERR					;SET LOCAL LOOP ON ERROR
4250	010650	010203				MOV	R2,R3					;BUILD POINTER TO RKDB
4251	010652	062703	000024			ADD	#RKDB,R3					
4252	010656	012701	070414			MOV	#IBUFF,R1					;SET POINTER TO BUFFER
4253	010662	004437	036252			JSR	R4,LRLD					;LOAD "L" REGS
4254	010666	000125				RDHEAD						;READ HEADER
4255	010670	000000				0						;NO WORDS COUNT
4256	010672	000000				0						;NO BUFFER
4257	010674	000				.BYTE	0					;SECTOR 0
4258	010675	000			10\$:	.BYTE	0					; TRACK 0
4259	010676	000312				312						; CYL 312
4260												
4261	010700	000402				BR	2\$;SKIP INIT
4262	010702				112\$:							
4263	010702	104416				TSSINIT						;CLEAR SUBSYSTEM
4264	010704	104003				ERROR	3					;BAD INIT ERROR
4265	010706	104417			2\$:	TLOADRK						;LOAD RK REGS
4266	010710	104423				TWAT16						;WAIT FOR INTERRUPT
4267	010712	104002				ERROR	2					;TO SLOW/NOT COMPLETE ERROR
4268												
4269	010714	104421				TCHKOP						;CHECK FOR ANY ERRORS
4270	010716	104004				ERROR	4; OR 5,6,7					;REPORT ALL ERRORS
4271	010720	012700	000003			MOV	#3,R0					;SET COUNT
4272	010724	011321			5\$:	MOV	(R3),(R1)+					;GET RKDB
4273	010726	104420				TGETRK						;GET RK REGS
4274	010730	032737	100000	001550		BIT	#DLT,T.CS2					;TEST IF DATA LATE
4275	010736	001410				BEQ	3\$;NO-SKIP
4276	010740	012737	055173	054622		MOV	#EMDLT,EM13					
4277	010746	012737	057341	061454		MOV	#EMRDB,DF011A					
4278	010754	104013				ERROR	13					; "DATA LATE SET RESULT OF DB READ
4279	010756	104415				SCOPI						; LOCAL LOOP TO 112\$
4280												
4281	010760	032737	100000	001540	3\$:	BIT	#CERR,T.CS1					;TEST IF CONT ERROR SET
4282	010766	001410				BEQ	4\$;NO-SKIP
4283	010770	012737	057365	001500		MOV	#EMCERR,EM13N					
4284	010776	012737	057341	061454		MOV	#EMRDB,DF011A					
4285	011004	104013				ERROR	13					; "CERR SET RESULT OF READ DB
4286	011006	104415				SCOPI						; LOCAL LOOP TO 112\$
4287	011010	005300			4\$:	DEC	R0					;DEC COUNT
4288	011012	001344				BNE	5\$;LOOP IF NOT ZERO
4289	011014	012737	011024	001110		MOV	#117\$,SLPERR					;SET LOCAL LOOP 117\$
4290	011022	000402				BR	7\$;SKIP INIT
4291	011024				117\$:							
4292	011024	104416				TSSINIT						;CLEAR SUBSYSTEM
4293	011026	104003				ERROR	3					;BAD INIT ERROR
4294	011030	004437	036740		7\$:	JSR	R4,RDSTHD					;GO READ & SEQUENCE HEADERS

```

4295 011034 104421          TCHKOP          ;CONTROLLER ERROR RETURN
4296 011036 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4297 011040 104013          ERROR 13 ;"DATA LATE GET RESULT OF DATA BUFFER READ"
4298 011042 104002          ERROR 2 ;"OPERATION TO SLOW" MESSAGE
4299                                     ;OR "HEADER 0 NOT FOUND" MESSAGE
4300
4301 011044 004437 042636    JSR R4,GENCOM
4302 011050 100200          100200          ;COMPARE IBUF & OBUF (HEADERS)
4303 011052 000414          BR 6$           ;GOOD RETURN-NO MISCOMPARES
4304 011054 104015          ERROR 15        ;REPORT 1ST MSCOMPARES
4305
4306 011056 013700 001634    MOV ERRHLT,RO   ;GET ERROR LIMIT
4307 011062 005300          DEC RO          ;DECREMENT IT
4308 011064 001407          BEQ 6$         ;EXIT IF ZERO
4309 011066 004437 042636    JSR R4,GENCOM
4310
4311 011072 040000          040000          ;RESUME COMPARE
4312 011074 000403          BR 6$           ;GOOD RETURN-NO MORE ERRORS
4313 011076 104016          ERROR 16        ;REPORT NEXT ERROR LINE
4314 011100 000770          BR 12$         ;LOOP
4315 011102 104415          SCOP1          ;LOCAL ERROR LOOP TO 117$
4316
4317 011104 105737 001607    6$: TSTB L.DT    ;WAS TRACK 1 JUST DONE?
4318 011110 001010          BNE 8$         ;YES-SKIP
4319
4320 011112 112737 000001 010601 MOVB #1,11$    ;CHANGE PARAM TO LOAD "L" WITH
4321 011120 112737 000001 010675 MOVB #1,10$    ;TRACK 2
4322 011126 000137 010566    JMP 9$         ;JUMP TO DO ENTIRE TEST ON TRK 1
4323
4324 011132          8$:
4325
4326          .SBTTL **HEADER RECOGNITION TESTS
4327
4328          ;*****
4329          ;*TEST 22          BAD SECTOR ERROR
4330          ;*
4331          ;*          FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4332          ;*          SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
4333          ;*          (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
4334          ;*          AND ALL OTHER SECTORS GOOD.
4335          ;*
4336          ;*          ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
4337          ;*          SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS. ISSUE A
4338          ;*          WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
4339          ;*          MAKE SURE BAD SECTOR ERROR SET. ISSUE A WRITE DATA
4340          ;*          OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2. MAKE
4341          ;*          SURE NO ERROR SETS.
4342          ;*
4343          ;*****
4344          ;*ST22: SCOPE
4345          011132 000004          MOV #10,$TIMES ;DO 10. ITERATIONS
4346          011134 012737 000012 001262 MOV #312,REFMT ;SET REFORMAT SWITCH
4347          011142 012737 000312 001676 TSSINIT        ;CLEAR SUBSYSTEM
4348          011150 104416          ERROR 3        ;BAD INIT ERROR
4349
4350 011154 004437 036252    JSR R4,LRLOAD  ;LOAD "L" REGS

```

4351	011160	000127				WRHEAD			;WRITE HEADER
4352	011162	177676				-102			;WORD COUNT FOR 26 SECTOR MODE
4353	011164	072414				OBUFF			;BUFFER ADDRESS
4354	011166	000				.BYTE	0		;SECTOR
4355	011167	000				.BYTE	0		;TRACK
4356	011170	000312				312			;CYLINDER
4357									
4358									
4359	011172	004437	042636			JSR	R4,GENCOM		;GENERATE HEADERS
4360	011176	000600				600			;WITH NO BS BITS
4361									
4362	011200	012700	072416			MOV	#OBUFF+2,RO		;RESET BIT 15 IN WORD 2 OF
4363	011204	042720	100000			BIC	#BIT15,(RO)+		;SECTOR 0 HEADER AND BIT 14
4364	011210	042720	100000			BIC	#BIT15,(RO)+		;IN WORD 2 OF SECTOR 1 HEADER.
4365	011214	005720				TST	(RO)+		;ALSO CORRECT THE VRC
4366	011216	042720	040000			BIC	#BIT14,(RO)+		
4367	011222	042710	040000			BIC	#BIT14,(RO)		
4368									
4369	011226	104417				TLOADRK			;LOAD RK REGS
4370	011230	104431				TWAT112			;WAIT FOR INTERRUPT
4371	011232	104002				ERROR	2		;TO SLOW/NOT COMPLETE ERROR
4372									
4373	011234	104421				TCHKOP			;CHECK IF ANY ERRORS
4374	011236	104004				ERROR	4 ; OR 5,6,7		;REPORT ALL ERRORS
4375	011240	012737	011246	001110		MOV	#4\$,\$LPERR		;SET LOCAL LOOP ON ERROR
4376	011246	104416			4\$:	TSSINIT			
4377	011250	104003				ERROR	3		
4378	011252	004437	036252			JSR	R4,LRLOAD		;LOAD "L" REGS
4379	011256	000123				WRDATA			;WRITE DATA
4380	011260	177400				-400			;WORD COUNT
4381	011262	072414				OBUFF			;BUS ADDRESS
4382	011264	000			5\$:	.BYTE	0		;SECT 0
4383	011265	000				.BYTE	0		;TRACK 0
4384	011266	000312				312			;CYL 312
4385									
4386	011270	104417			1\$:	TLOADRK			;LOAD RK REGS
4387	011272	104424				TWAT32			;WAIT FOR INTERRUPT
4388	011274	104002				ERROR	2		;TO SLOW/NOT COMPLETE ERROR
4389									
4390	011276	022737	000002	011264		CMP	#2,5\$;JUST READ SECTOR 2?
4391	011304	001415				BEG	6\$;YES - SKIP
4392									
4393	011306	104422				TCHKWE			;CHECK OPERATION WITH ERROR
4394	011310	000000				0			
4395	011312	000100				100			;EXPECTED BSE
4396	011314	000000				0			
4397	011316	104004				ERROR	4 ; OR 5,6,7		;REPORT ANY DISCREPENCIES
4398									
4399	011320	104415				SCOPI			;LOCAL ERROR LOOP TO 4\$
4400									
4401	011322	122737	000002	011264		CMPB	#2,5\$;WAS SECTOR SET TO 2
4402	011330	001405				BEG	7\$;YES-SKIP
4403	011332	105237	011264			INCB	5\$;BUMP TO NEXT SECTOR
4404	011336	000743				BR	4\$;LOOP
4405									
4406	011340	104421			6\$:	TCHKOP			;CHECK FOR GOOD OPERATION

ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS

4407 011342 104004
4408
4409 011344
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420 011344 000004
4421 011346 012737 000012 001262
4422 011354 012737 000312 001676
4423 011362 104416
4424 011364 104003
4425
4426 011366 004437 036252
4427 011372 000127
4428 011374 177676
4429 011376 072414
4430 011400 000
4431 011401 000
4432 011402 000312
4433
4434 011404 004437 042636
4435 011410 000600
4436
4437 011412 012700 072420
4438 011416 012703 000001
4439 011422 030310
4440 011424 001402
4441 011426 040310
4442 011430 000401
4443 011432 050310
4444 011434 062700 000006
4445
4446 011440 006303
4447 011442 001367
4448
4449 011444 104417
4450 011446 104431
4451 011450 104002
4452 011452 104421
4453 011454 104004
4454
4455 011456 012737 011464 001110
4456 011464 104416
4457 011466 104003
4458
4459 011470 004437 036252
4460 011474 000123
4461 011476 177400
4462 011500 072414

```

7$:
*****
*TEST 23      HEADER VRC ERROR
*****
*
*   FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
*   16 SECTORS WITH BAD HEADER VRC.  ISSUE A WRITE DATA
*   OF EACH OF THE SECTORS WITH A BAD HEADER VRC.  MAKE
*   SURE HEADER VRC ERROR SETS.  ISSUE A WRITE DATA TO
*   A GOOD HEADER AND MAKE SURE NO ERROR OCCURS.
*****
†ST23:
SCOPE
MOV      #10, $TIMES      ;DO 10. ITERATIONS
MOV      #312, REFMT     ;SET REFORMAT SWITCH
TSSINIT                      ;CLEAR SUBSYSTEM
ERROR    3                ;BAD INIT ERROR
JSR      R4, LRLOAD      ;LOAD "L" REGS
WRHEAD                      ;WRITE HEADER
-102                          ;WORD COUNT
OBUFF                      ;BUFF ADD
.BYTE    0                 ;SECT
.BYTE    0                 ;TRACK
312                          ;CYL
JSR      R4, GENCOM
600                          ;BUILD HEADERS NO BSE
MOV      #OBUFF+4, R0     ;GET ADDRESS OF VRC HDR0
MOV      #BIT0, R3        ;SET FOR BIT CHANGE SELECT
1$:    BIT      R3, (R0)   ;CHECK A VRC BIT
      BEQ      2$,        ;SKIP IF ZERO
      BIC      R3, (R0)   ;ELSE CLEAR IT
      BR       3$,        ;SKIP
2$:    BIS      R3, (R0)   ;IF ZERO SET IT
3$:    ADD      #6, R0     ;BUMP TO NEXT VRC WORD
ASL      R3                ;SHIFT THE SELECT
BNE      1$               ;IF BIT NOT SHIFTED OUT-LOOP
TLOADRK                      ;LOAD RK REGS
TWAT112                      ;WAIT FOR INTERRUPT
ERROR    2                ;TO SLOW/NOT COMPLETE ERROR
TCHKOP                      ;CHECK OPERATION COMPLETE
ERROR    4 ; OR 5,6,7    ;REPORT ALL ERRORS
MOV      #4$, $LPERR      ;SET LOCAL LOOP
TSSINIT                      ;CLEAR SUBSYSTEM
ERROR    3                ;BAD INIT ERROR
JSR      R4, LRLOAD      ;LOAD "L" REGS
WRDATA                      ;WRITE DATA
-400                          ;WORD COUNT
OBUFF                      ;BUFFER ADD

```

```

4463 011502 000          5$: .BYTE 0 ;SECT
4464 011503 000          .BYTE 0 ;TRACK
4465 011504 000312      312 ;CYL
4466
4467 011506 104417      TLOADRK ;LOAD RK REG
4468 011510 104424      TWAT32 ;WAIT FOR INTERRUPT
4469 011512 104002      ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4470
4471 011514 022737 000020 011502  CMP #16.,5$ ;WAS THIS WRITE SECTOR 16?
4472 011522 001415      BEQ 6$ ;YES-SKIP
4473
4474 011524 104422      TCHKWE ;CHECK OPERATION WITH ERROR
4475 011526 000000      0
4476 011530 000040      40 ;HVRC EM EXPECTED
4477 011532 000000      0
4478 011534 104004      ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
4479
4480 011536 104415      SCOPI ;LOCAL LOOP TO 4$
4481
4482 011540 105237 011502      INCB 5$ ;BUMP SECTOR IN "L" REG
4483 011544 022737 000016 011502  CMP #16,5$ ;IF SECTOR IS 16 OR LESS
4484 011552 003744      BLE 4$ ;LOOP
4485 011554 000402      BR 7$ ;ELSE EXIT
4486 011556 104421      TCHKOP ;CHECK LAST OPERATION NO ERRORS
4487 011560 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4488
4489 011562
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499 011562 000004          *****
4500 011564 012737 000012 001262  †ST24: SCOPE ;DO 10. ITERATIONS
4501 011572 012737 000312 001676  MOV #10.,$TIMES ;SET REFORMAT SWITCH
4502 011600 104416      TSSINIT ;CLEAR SUBSYSTEM
4503 011602 104003      ERROR 3 ;BAD INIT ERROR
4504
4505 011604 004437 036252      JSR R4,LRLOAD ;LOAD "L" REG
4506 011610 000127      WRHEAD ;WRITE HEADER
4507 011612 177676      -102 ;WORD CNT FOR 26 SECTOR MODE
4508 011614 072414      OBUFF ;BUFF ADD
4509 011616 000          .BYTE 0 ;SECTOR
4510 011617 000          .BYTE 0 ;TRACK
4511 011620 000312      312 ;CYLINDER
4512
4513 011622 004437 042636      JSR R4,GENCOM ;BUILD HEADERS-NO BSE
4514 011626 000600      600
4515
4516 011630 042737 100000 072416  BIC #BIT15,OBUFF+2 ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
4517
4518 011636 104417      TLOADRK ;LOAD RK REGS

```

```

4519 011640 104431          TWAT112          ;WAIT FOR INTERRUPT
4520 011642 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
4521
4522 011644 104421          TCHKOP          ;CHECK FOR ANY ERRORS
4523 011646 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4524
4525 011650 004437 036252    JSR R4,LRLOAD   ;LOAD "L" REGS
4526 011654 000123          WRDATA          ;WRITE DATA
4527 011656 177400          -400           ;WORD COUNT
4528 011660 072414          OBUFF          ;BUFF ADD
4529 011662 000           .BYTE 0        ;SECTOR
4530 011663 000           .BYTE 0        ;TRACK
4531 011664 000312          312           ;CYLINDER
4532
4533 011666 104417          TLOADRK        ;LOAD RK REGS
4534 011670 104424          TWAT32        ;WAIT FOR INTERRUPT
4535 011672 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
4536
4537 011674 104422          TCHKWE        ;CHECK OPERATION WITH EXPECTED ERR
4538 011676 000000          0             ;
4539 011700 000040          40            ;HVRC ERR EXPECTED
4540 011702 000000          0             ;
4541 011704 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL DISCREPENCIES
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551 011706 000004          *TEST 25      OPERATION INCOMPLETE
4552 011710 012737 000012 001262
4553 011716 012737 000312 001676
4554 011724 104416          *
4555 011726 104003          *
4556
4557 011730 004437 036252    JSR R4,LRLOAD   ;LOAD "L" REGS
4558 011734 000127          WRHEAD        ;WRITE HEADER
4559 011736 177676          -102         ;WORD COUNT FOR 26 SECT MODE
4560 011740 072414          OBUFF        ;BUFF ADD
4561 011742 000           .BYTE 0        ;SECTOR
4562 011743 000           .BYTE 0        ;TRACK
4563 011744 000312          312         ;CYLINDER
4564
4565 011746 004437 042636    JSR R4,GENCOM   ;BUILD HEADERS-NO BSE ERRORS
4566 011752 000600          600
4567
4568 011754 052737 001000 072614    BIS #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
4569 011762 052737 001000 072616    BIS #BIT9,OBUFF+202 ;CORRECT THE VRC
4570
4571 011770 104417          TLOADRK        ;LOAD RK REGS
4572 011772 104431          TWAT112        ;WAIT FOR INTERRUPT
4573 011774 104002          ERROR 2        ;TO SLOW/NOT COMPLETE
4574

```



```

4575 011776 104421 TCHKOP ;CHECK FOR ANY ERRORS
4576 012000 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4577
4578 012002 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
4579 012006 000123 WRDATA ;WRITE DATA
4580 012010 177400 -400 ;400 WORDS
4581 012012 072414 OBUFF ;BUFF ADD
4582 012014 025 .BYTE 25 ;SECTOR 25
4583 012015 000 .BYTE 0 ;TRACK 0
4584 012016 000312 312 ;CYL 312
4585
4586 012020 104417 TLOADRK ;LOAD RK REGS
4587 012022 104425 TWAT48 ;WAIT FOR INTERRUPT
4588 012024 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
4589
4590 012026 104422 TCHKWE ;CHECK OPERATION EXPECTED ERROR
4591 012030 000000 0
4592 012032 000020 20 ;OPI EXPECTED
4593 012034 000000 0
4594 012036 104004 ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPANCIES

```

```

*****
*TEST 26 OPI WITH HVRC ERROR
*
* FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
* ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
* ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
* TRACK 0, SECTOR 17. THAT BOTH OPERATION INCOMPLETE
* AND HEADER VRC SET.
*
*****

```

```

4605 012040 000004 ST26: SCOPE
4606 012042 012737 000012 001262 MOV #10,STIMES ;DO 10. ITERATIONS
4607 012050 012737 000312 001676 MOV #312,REFMT ;SET REFORMAT SWITCH
4608 012056 104416 TSSINIT ;CLEAR SUBSYSTEM
4609 012060 104003 ERROR 3 ;BAD INIT ERROR
4610
4611 012062 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
4612 012066 000127 WRHEAD ;WRITE HEADER
4613 012070 177676 -102 ;WORD COUNT FOR 26 SECT MODE
4614 012072 072414 OBUFF ;BUS ADDRESS
4615 012074 000 .BYTE 0 ;SECTOR
4616 012075 000 .BYTE 0 ;TRACK
4617 012076 000312 312 ;CYLINDER
4618
4619 012100 004437 042636 JSR R4,GENCOM
4620 012104 000600 600 ;BUILD HEADER- NO BSE ERRORS
4621
4622 012106 012700 072550 MOV #OBUFF+134,R0 ;GET ADDRESS 2ND WORD HDR 17(8)
4623 012112 052720 001000 BIS #BIT9,(R0)+ ;SET FORMAT 24 SECT PER TRACK
4624 012116 052720 001000 BIS #BIT9,(R0)+ ;SET VRC BIT
4625 012122 062700 000004 ADD #4,R0 ;BUMP TO HVRC WORD HDR 20(8)
4626 012126 032710 000001 BIT #BIT0,(R0) ;TEST BIT 0
4627 012132 001403 BEQ 1$ ;RESET-SKIP
4628 012134 042710 000001 BIC #BIT0,(R0) ;CLEAR BIT
4629 012140 000402 BR 2$
4630 012142 052710 000001 1$: BIS #BIT0,(R0) ;SET BIT

```

20 46

```

4631                                ;FORCE OPI AND HVRC ERROR
4632 012146 104417                2$: TLOADRK ;LOAD RK REGS
4633 012150 104431                TWAT112 ;WAIT FOR INTERRUPT
4634 012152 104002                ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4635
4636 012154 104421                TCHKOP ;CHECK FOR ANY ERRORS
4637 012156 104004                ERROR 4 ; OR 5,6,7 ;YES-REPORT ALL ERRORS
4638
4639 012160 004437 036252        JSR R4,LRLOAD ;LOAD "L" REGS
4640 012164 000123                WRDATA ;WRITE DATA
4641 012166 177400                -400 ;400 WORDS
4642 012170 072414                OBUFF ;BUFF ADDRESS
4643 012172 017 ;SECT 17
4644 012173 000 ;TRACK 0
4645 012174 000312                312 ;CYLINDER 312
4646
4647 012176 104417                TLOADRK ;LOAD RK REGS
4648 012200 104425                TWAT48 ;WAIT FOR INTERRUPT
4649 012202 104002                ERROR 2 ;TO SLOW/NOT COMPLETE
4650
4651 012204 104422                TCHKWE ;CHECK WITH EXPECTED ERROR
4652 012206 000000                0
4653 012210 000060                60 ;HVRC ERR & OPI EXPECTED
4654 012212 000000                0
4655 012214 104004                ERROR 4 ;OR 5,6,7
4656 *****
4657 *TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
4658 *
4659 * FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
4660 * ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
4661 * AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
4662 * END OF THE OPERATION
4663 *
4664 *****
4665 *ST27: SCOPE
4666 012216 000004                MOV #10,STIMES ;DO 10. ITERATIONS
4667 012220 012737 000012 001262    MOV #312,REFMT ;SET REFORMAT SWITCH
4668 012226 012737 000312 001676
4669 012234 104416                TSSINIT ;CLEAR SUBSYSTEM
4670 012236 104003                ERROR 3 ;BAD INIT ERROR
4671
4672 012240 004437 036252        JSR R4,LRLOAD ;LOAD "L" REGISTERS
4673 012244 000127                WRHEAD ;WRITE HEADER
4674 012246 177676                -102 ;WORD COUNT FOR 26 SECTOR MODE
4675 012250 072414                OBUFF ;BUFF ADD
4676 012252 000 ;SECTOR
4677 012253 000 ;TRACK
4678 012254 000312                312 ;CYLINDER
4679
4680 012256 004437 042636        JSR R4,GENCOM ;BUILD HEADERS-NO BSE ERRORS
4681 012262 000600
4682
4683 012264 012700 072560        MOV #OBUFF+144,R0 ;ADDRESS OF HEAD 20 HVRC WORD
4684 012270 012701 000002        MOV #BIT1,R1 ;BIT 1 CONSTANT
4685 012274 030110                BIT R1,(R0) ;TEST BIT 1 SET
4686 012276 001402                BEQ 1$ ;RESET-SKIP

```

```

4687 012300 040110      BIC      R1,(R0)      ;ELSE CLEAR BIT 1
4688 012302 000401      BR       2$           ;SKIP
4689 012304 050110      1$:     BIS      R1,(R0) ;SET BIT 1
4690
4691 012306 104417      2$:     TLOADRK      ;LOAD RK REGS
4692 012310 104431      TWAT112 ;WAIT FOR INTERRUPT
4693 012312 104002      ERROR 2 ;TO SLOW/NOT COMPLETE
4694
4695 012314 104421      TCHKOP      ;CHECK FOR ANY ERROR
4696 012316 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4697
4698 012320 004437 036252 JSR      R4,LRLOAD   ;LOAD "L" REGISTER
4699 012324 000123      WRDATA      ;WRITE DATA
4700 012326 177400      -400        ;WORD COUNT
4701 012330 072414      OBUFF       ;BUFF ADD
4702 012332 021         .BYTE 21        ;SECTOR
4703 012333 000         .BYTE 0         ;TRACK
4704 012334 000312      312        ;CYLINDER
4705 012336 104417      TLOADRK      ;LOAD RK REGS
4706 012340 104424      TWAT32      ;WAIT FOR INTERRUPT
4707 012342 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
4708
4709 012344 104421      TCHKOP      ;CHECK FOR ANY ERROR
4710 012346 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS.

```

.SBTTL **DATA TRANSFER TESTS

```

*****
*TEST 30      WRITE AND READ ONE SECTOR
*****
*
*      FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
*      TO AGREED WITH BAD SECTOR INFORMATION.  ISSUE A WRITE DATA
*      OF ONE SECTOR ON CYLINDER 312, TRACK 0.  READ IT BACK TO
*      MAKE SURE IT AGREES WITH WHAT IS WRITTEN.
*****

```

```

4723
4724 012350 000004      †ST30:  SCOPE
4725 012352 012737 000012 001262  MOV      #10,$TIMES ;DO 10. ITERATIONS
4726 012360 012737 000312 001676  MOV      #312,REFMT ;SET REFORMAT SWITCH
4727 012366 104416      TSSINIT      ;CLEAR SUBSYSTEM
4728 012370 104003      ERROR 3      ;BAD INIT ERROR
4729
4730 012372 012737 000312 012420  MOV      #312,7$   ;PRESET CYL POINTER
4731 012400 105037 012417      CLRB 2$        ;CLEAR TRACK POINTER
4732
4733 012404 004437 036252  1$:     JSR      R4,LRLOAD   ;LOAD "L" REG
4734 012410 000127      WRHEAD      ;WRITE HEADER
4735 012412 177676      -102        ;WORD COUNT FOR 26 SECTOR MODE
4736 012414 072414      OBUFF       ;BUFF ADDRESS
4737 012416 000         .BYTE 0         ;SECTOR
4738 012417 000         .BYTE 0         ;TRACK
4739 012420 000312      7$:     312      ;CYLINDER
4740
4741 012422 004437 042636 JSR      R4,GENCOM
4742 012426 001200      1200      ;BUILD HDRS-INCLUDE BAD SECTORS

```

4743										
4744	012430	104417				TLOADRK				;LOAD RK REGS
4745	012432	104431				TWAT112				;WAIT FOR INTERRUPT
4746	012434	104002				ERROR	2			;TO SLOW/NOT COMPLETE ERROR
4747										
4748	012436	104421				TCHKOP				;CHECK FOR ANY ERRORS
4749	012440	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS
4750										
4751	012442	022737	000313	012420		CMP	#313,7\$;TEST IF DONE 313 TK 0
4752	012450	001414				BEQ	3\$;YES - SKIP
4753	012452	123727	012417	000002		CMPB	2\$, #2			;DID WE JUST FORMAT TRACK 2
4754	012460	001403				BEQ	8\$;YES-SKIP
4755	012462	105237	012417			INCB	2\$;BUMP TO NEXT TRACK
4756	012466	000746				BR	1\$;GO FORMAT NEXT TRACK
4757										
4758	012470	105037	012417		8\$:	CLRB	2\$;CLEAR TRACK POINTER
4759	012474	005237	012420			INC	7\$;BUMP CYL TO 313
4760	012500	000741				BR	1\$;GO FORMAT 313 TK 0
4761										
4762	012502	004437	036252		3\$:	JSR	R4, LRLOAD			;LOAD "L" REGS
4763	012506	000123				WRDATA				;WRITE DATA
4764	012510	177400				-400				;ONE SECTOR WORD COUNT
4765	012512	072414				OBUFF				;BUFF ADDRESS
4766	012514	012				.BYTE	12			;SECTOR 12
4767	012515	000				.BYTE	0			;TRACK 0
4768	012516	000312				312				;CYLINDER 312
4769										
4770	012520	004437	042636			JSR	R4, GENCOM			
4771	012524	000001				1				;BUILD DATA PATTERN 1
4772	012526	000400				400				;400 WORDS LONG
4773	012530	012737	012536	001110		MOV	#4\$, \$LPERR			;SET FOR LOCAL LOOP
4774	012536	104417			4\$:	TLOADRK				;LOAD RK REGS
4775	012540	104431				TWAT112				;WAIT FOR INTERRUPT
4776	012542	104002				ERROR	2			;TO SLOW/NOT COMPLETE ERROR
4777										
4778	012544	104421				TCHKOP				;CHECK FOR ANY ERRORS
4779	012546	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS
4780										
4781	012550	004437	036252			JSR	R4, LRLOAD			;LOAD "L" REGS
4782	012554	000121				RDDATA				;READ DATA
4783	012556	177400				-400				;400 WORDS
4784	012560	070414				IBUFF				;BUFF ADD
4785	012562	012				.BYTE	12			;SECTOR 12
4786	012563	000				.BYTE	0			;TRACK 0
4787	012564	000312				312				;CYL 312
4788										
4789	012566	104417				TLOADRK				;LOAD RK
4790	012570	104424				TWAT32				;WAIT FOR INTERRUPT
4791	012572	104002				ERROR	2			;TO SLOW/NOT COMPLETE
4792										
4793	012574	104421				TCHKOP				;CHECK FOR ANY ERRORS
4794	012576	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS
4795										
4796	012600	004437	042636			JSR	R4, GENCOM			
4797	012604	100001				100001				;GO COMPARE DATA TO PATTERN 1
4798	012606	000400				400				;400 WORDS LONG

```

4799 012610 000413 BR 6$ ;GOOD RETURN-NO DATA ERRORS
4800 012612 104015 ERROR 15 ;ERROR RETURN
4801
4802 012614 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
4803 012620 005300 5$: DEC R0 ;DEC LIMIT
4804 012622 001406 BEQ 6$ ;EXIT IF 0
4805 012624 004437 042636 JSR R4,GENCOM
4806 012630 040000 040000 ;CONTINUE COMPARE
4807 012632 000402 BR 6$ ;EXIT IF NO MORE ERRORS
4808 012634 104016 ERROR 16 ;ELSE REPORT MISCOMPARE
4809 012636 000770 BR 5$ ;LOOP
4810 012640 005037 001676 6$: CLR REFORMAT ;CLEAR REFORMAT SWITCH
4811
4812 ;*****
4813 ;*TEST 31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT
4814 ;*
4815 ;* ISSUE A WRITE DATA OF ONE SECTOR TO CYLINDER 312,
4816 ;* TRACK 2, SECTOR 12 WITH INHIBIT BUS
4817 ;* ADDRESS INCREMENT. READ DATA BACK TO MAKE SURE
4818 ;* EVERY WORD IS THE SAME AND CORRECT.
4819 ;*
4820 ;*****
4821 012644 000004 TST31: SCOPE
4822 012646 012737 000012 001262 MOV #10.,$TIMES ;;DO 10. ITERATIONS
4823
4824 012654 104416 TSSINIT ;CLEAR SUBSYSTEM
4825 012656 104003 ERROR 3 ;BAD INIT ERROR
4826
4827 012660 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
4828 012664 000123 WRDATA ;WRDATA
4829 012666 177400 -400 ;-400 WORDS
4830 012670 072414 OBUFF ;OBUFF IS BUFF ADDRESS
4831 012672 012 .BYTE 12 ;SECTOR 12
4832 012673 001 .BYTE 1 ;TRACK 1
4833 012674 000312 312 ;CYLINDER 312
4834
4835 012676 052737 000020 001610 BIS #BAI.L.CS2 ;SET INCREMENT INHIBIT
4836 012704 004437 042636 JSR R4,GENCOM ;BUILD PATTERN
4837 012710 000016 16 ;PATTERN 16
4838 012712 000400 400 ;400 WORDS
4839
4840 012714 104417 TLOADRK ;LOAD RK REGS
4841 012716 104430 TWAT96 ;WAIT FOR INTERRUPT
4842 012720 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4843
4844 012722 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4845 012724 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4846
4847 012726 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
4848 012732 000121 RDDATA ;RDDATA
4849 012734 177400 -400 ;-400 WORDS
4850 012736 070414 IBUFF ;IBUFF IS BUFF ADDRESS
4851 012740 012 .BYTE 12 ;SECTOR 12
4852 012741 001 .BYTE 1 ;TRACK 1
4853 012742 000312 312 ;CYLINDER 312
4854

```

```

4855 012744 012700 000377      MOV      #377,R0      ;SET COUNT TO SET OBUFF TO BE
4856 012750 012701 072416      MOV      #0BUFF+2,R1 ;ALL THE FIRST WORD OF PATTERN
4857 012754 012703 072414      MOV      #0BUFF,P3
4858
4859 012760 011321      1$:      MOV      (R3),(R1)+  ;MOV THE WORD
4860 012762 005300      DEC      R0
4861 012764 001375      BNE     1$           ;LOOP UNTIL ALL WORDS SET
4862
4863 012766 104417      TLOADRK          ;LOAD RK REGS
4864 012770 104424      TWAT32          ;WAIT FOR INTERRUPT
4865 012772 104002      ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
4866
4867 012774 104421      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
4868 012776 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4869
4870 013000 004437 042636      JSR      R4,GENCOM ;COMPARE THE DATA
4871 013004 100000
4872 013006 000400      400
4873 013010 000413      BR      2$
4874 013012 104015      ERROR 15
4875 013014 013700 001634      MOV      ERRLMT,R0  ;GET ERROR LIMIT
4876 013020 005300      64$:      DEC      R0         ;DECREMENT COUNT
4877 013022 001406      BEQ     65$        ;IF ZERO - EXIT
4878 013024 004437 042636      JSR      R4,GENCOM ;CONTINUE DATA COMPARE
4879 013030 040000      40000
4880 013032 000402      BR      65$        ;NO MORE ERRORS - EXIT
4881 013034 104016      ERROR 16         ;REPORT NEXT ERROR
4882 013036 000770      BR      64$        ;LOOP
4883 013040      65$:
4884
4885 013040      2$:
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896 013040 000004      *****
4897 013042 012737 000012 001262      *TEST 32 WRITE DATA ADDRESS GREATER THAN 32K
4898 013050 123727 001326 000001      *
4899 013056 002016      *
4900
4901 013060 032737 000002 001664      * ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
4902 013066 001011      * MAKE SURE CORRECT DATA IS ON DISK.
4903
4904 013070 104401 052135      *
4905 013074 104401 052300      * NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
4906 013100 052737 000002 001664      * OF MEMORY IS PRESENT.
4907 013106 104401 052314      *
4908 013112 000470      *
4909 013114 012737 013122 001110      *****
4910 013122      *

```

```

↑ST32: SCOPE
MOV      #10,$TIMES      ;;DO 10. ITERATIONS
CMPB    $MAMS1,#1       ;TEST IF >32K MEM
BGE     2$              ;YES-SKIP
BIT      #MEMSZB,OPTFLG ;TEST IF REPORT ALREADY MADE
BNE     1$              ;YES -SKIP
TYPE    ,OPR011         ;"INSUFFICIENT MEMORY DATA TRANSFER WITH
TYPE    ,OPR012         ;ADDRESS >32K
BIS     #MEMSZB,OPTFLG ;SET FLAG
TYPE    ,OPR015         ;"BYPASSED"
BR      4$              ;EXIT
MOV     #5$,$LPERR      ;SET LOCAL LOOP ON ERROR ADDRESS

```

4911	013122	104416		TSSINIT		;CLEAR SUBSYSTEM
4912	013124	104003		ERROR 3		;BAD INIT ERROR
4913						
4914	013126	004437	036252	JSR R4,LRLOAD		;LOAD "L" REGS
4915	013132	000123		WRDATA		;WRITE DATA
4916	013134	177400		-400		;400 WORDS
4917	013136	177770		177770		;BUS ADDRESS IN 32K -10 BYTES
4918	013140	016		.BYTE 16		;SECTOR 16
4919	013141	000		.BYTE 0		;TRACK 0
4920	013142	000312		312		;CYLINDER 312
4921	013144	004437	042636	JSR R4,GENCOM		;GENERATE DATA
4922	013150	010010		10010		;PATTERN 10, MEM. MANAGEMENT FOR DEST.
4923	013152	001777		1777		;RELOCATION ARGUMENT
4924	013154	000400		400		;400 WORDS
4925						
4926	013156	104417		TLOADRK		;LOAD RK REGS
4927	013160	104430		TWAT96		;WAIT FOR INTERRUPT
4928	013162	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
4929						
4930	013164	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
4931	013166	104004		ERROR 4 ;OR 5, 6, 7,	10	;REPORT ALL ERRORS
4932	013170	104415		SCOP1		;LOCAL LOOP ON ERROR TO 5\$
4933						
4934	013172	004437	042636	JSR R4,GENCOM		;CLEAR Ibuff TO 1'S.
4935	013176	002007		2007		
4936	013200	001000		1000		
4937						
4938	013202	004437	036252	JSR R4,LRLOAD		;LOAD "L" REGS
4939	013206	000121		RDDATA		;RDDATA
4940	013210	177400		-400		;400 WORDS
4941	013212	070414		IBUFF		;IBUFF IS BUFF ADDRESS
4942	013214	016		.BYTE 16		;SECTOR 16
4943	013215	000		.BYTE 0		;TRACK 0
4944	013216	000312		312		;CYLINDER 312
4945	013220	104417		TLOADRK		;LOAD RK REGS
4946	013222	104424		TWAT32		;WAIT FOR INTERRUPT
4947	013224	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
4948	013226	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
4949	013230	104004		ERROR 4 ;OR 5, 6, 7,	10	;REPORT ALL ERRORS
4950	013232	004437	042636	JSR R4,GENCOM		;COMPARE
4951	013236	110000		110000		;MEMORY MANAGEMENT FOR DESTINATION
4952	013240	001777		1777		;RELOCATION ARGUMENT
4953	013242	000400		400		;400 WORDS
4954	013244	000413		BR 4\$;NO ERROR-SKIP
4955	013246	104015		ERROR 15		;REPORT FIRST MISCOMPARE
4956	013250	013700	001634	MOV ERRMT,RO		;GET ERROR LIMIT
4957	013254	005300		DEC RO		;DECREMENT COUNT
4958	013256	001406		BEQ 65\$;IF ZERO - EXIT
4959	013260	004437	042636	JSR R4,GENCOM		;CONTINUE DATA COMPARE
4960	013264	050000		50000		
4961	013266	000402		BR 65\$;NO MORE ERRORS - EXIT
4962	013270	104016		ERROR 16		;REPORT NEXT ERROR
4963	013272	000770		BR 64\$;LOOP
4964	013274					
4965						
4966	013274					

```

4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977 013274 000004
4978 013276 012737 000012 001262
4979 013304 123727 001326 000001
4980 013312 002001
4981
4982 013314 000462 1$: BR 5$ ;EXIT
4983
4984 013316 012737 013324 001110 2$: MOV #3$, $LPERR ;SET LOCAL ERROR LOOP
4985
4986 013324 3$:
4987 013324 104416 TSSINIT ;CLEAR SUBSYSTEM
4988 013326 104003 ERROR 3 ;BAD INIT ERROR
4989 013330 004437 036252 JSR R4, LRLOAD ;LOAD "L" REGS
4990 013334 000123 WRDATA ;WRDATA
4991 013336 177400 -400 ;-400 WORDS
4992 013340 072414 OBUFF ;OBUFF IS BUFF ADDRESS
4993 013342 017 .BYTE 17 ;SECTOR 17
4994 013343 000 .BYTE 0 ;TRACK 0
4995 013344 000312 312 ;CYLINDER 312
4996 013346 004437 042636 JSR R4, GENCOM ;GENERATE DATA IN OBUFF
4997 013352 000011 11 ;PATTERN 11
4998 013354 000400 400 ;400 WORDS
4999
5000 013356 104417 TLOADRK ;LOAD RK REGS
5001 013360 104430 TWAT96 ;WAIT FOR INTERRUPT
5002 013362 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5003
5004 013364 104421 TCHKOP
5005 013366 104004 ERROR 4 ;OR 5, 6, 7, 10 ;CHECK OPERATION FOR ANY ERRORS
5006 013370 004437 036252 JSR R4, LRLOAD ;LOAD "L" REG
5007 013374 000121 RDDATA ;READ DATA
5008 013376 177400 -400 ;400 WORDS
5009 013400 177770 177770 ;ACROSS 32K BOUNDARY
5010 013402 017 .BYTE 17 ;SECTOR 17
5011 013403 000 .BYTE 0 ;TRACK 0
5012 013404 000312 312 ;CYL 312
5013
5014 013406 104417 TLOADRK ;LOAD RK REGS
5015 013410 104424 TWAT32 ;WAIT FOR INTERRUPT
5016 013412 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5017 013414 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5018 013416 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5019 013420 004437 042636 JSR R4, GENCOM ;COMPARE DATA
5020 013424 120000 120000 ;MEMORY MANAGEMENT WITH SOURCE
5021 013426 001777 1777 ;RELOCATION ARGUMENT
5022 013430 000400 400 ;COMPARE 400 WORDS

```

```

*****
*TEST 33 READ DATA ADDRESS GREATER THAN 32K
*
* ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 177770.
* CHECK MEMORY FOR CORRECT TRANSFER.
*
* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
* OF MEMORY IS PRESENT.
*****

```

```

*****
TST33: SCOPE
MOV #10, $TIMES ;DO 10. ITERATIONS
CMPB $MAMS1, #1 ;CHECK IF >32K MEMORY
BGE 2$ ;YES-SKIP
1$: BR 5$ ;EXIT
2$: MOV #3$, $LPERR ;SET LOCAL ERROR LOOP
3$:
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4, LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-400 ;-400 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312
JSR R4, GENCOM ;GENERATE DATA IN OBUFF
11 ;PATTERN 11
400 ;400 WORDS
TLOADRK ;LOAD RK REGS
TWAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP
ERROR 4 ;OR 5, 6, 7, 10 ;CHECK OPERATION FOR ANY ERRORS
JSR R4, LRLOAD ;LOAD "L" REG
RDDATA ;READ DATA
-400 ;400 WORDS
177770 ;ACROSS 32K BOUNDARY
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYL 312
TLOADRK ;LOAD RK REGS
TWAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
JSR R4, GENCOM ;COMPARE DATA
120000 ;MEMORY MANAGEMENT WITH SOURCE
1777 ;RELOCATION ARGUMENT
400 ;COMPARE 400 WORDS

```


E08

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 95
T33 READ DATA ADDRESS GREATER THAN 32K

SEQ 0095

```

5023 013432 000413 BR 5$ ;NO MISCOMPARE-EXIT
5024 013434 104015 ERROR 15 ;REPORT FIRST MISCOMPARE
5025 013436 013700 001634 MOV ERRMT,R0 ;GET ERROR LIMIT
5026 013442 005300 64$: DEC R0 ;DECREMENT COUNT
5027 013444 001406 BEQ 65$ ;IF ZERO - EXIT
5028 013446 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5029 013452 060000 60000
5030 013454 000402 BR 65$ ;NO MORE ERRORS - EXIT
5031 013456 104016 ERROR 16 ;REPORT NEXT ERROR
5032 013460 000770 BR 64$ ;LOOP
5033 013462
5034 013462
5035
5036 *****
5037 :TEST 34 WRITE DATA ADDRESS GREATER THAN 64K
5038 *
5039 * ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 37770.
5040 * MAKE SURE CORRECT DATA IS ON DISK.
5041 *
5042 * NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
5043 * OF MEMORY IS PRESENT.
5044 *****
5045 †ST34: SCOPE
5046 013462 000004 MOV #10,$TIMES ;DO 10. ITERATIONS
5047 013464 012737 000012 001262 CMPB $MAMS1,#2 ;CHECK IF >64K MEMORY
5048 013472 123727 001326 000002 BGE 2$ ;YES-SKIP
5049 013500 002016 6$: BIT $MEMSZB,OPTFLG ;TEST IF REPORT FLAG SET
5050 013502 032737 000002 001664 BNE 1$ ;NO-SKIP
5051 013510 001011
5052 013512 104401 052135 TYPE ,OPR011 ;"INSUFFICIENT MEMORY-DATA XFER WITH
5053 013516 104401 052304 TYPE ,OPR013 ;ADDRESS >64K
5054 013522 104401 052314 TYPE ,OPR015 ;"BYPASSED"
5055 013526 052737 000002 001664 BIS $MEMSZB,OPTFLG ;SET FLAG
5056 013534 000467 1$: BR 5$
5057
5058 013536 012737 013544 001110 2$: MOV #3,$LPERR ;SET LOCAL LOOP ON ERROR
5059
5060 013544 3$:
5061 013544 104416 TSSINIT ;CLEAR SUBSYSTEM
5062 013546 104003 ERROR 3 ;BAD INIT ERROR
5063 013550 004437 042636 JSR R4,GENCOM ;GENERATE DATA, PATTERN 11
5064 013554 010011 10011 ;MEM MANAGEMENT ON DESTINATION
5065 013556 003777 3777 ;RELOCATION ARGUMENT
5066 013560 000400 400 ;400 WORDS
5067
5068 013562 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5069 013566 000523 WRDATA!BA16 ;WRITE DATA AND SET BA16
5070 013570 177400 -400 ;400 WORDS
5071 013572 177770 177770 ;ACROSS 64K BOUNDARY
5072 013574 020 .BYTE 20 ;SECTOR 20
5073 013575 000 .BYTE 0 ;TRACK 0
5074 013576 000312 312 ;CYLINDER 312
5075
5076 013600 104417 TLOADRK ;LOAD RK REGS
5077 013602 104430 TWAT96 ;WAIT FOR INTERRUPT
5078 013604 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

```

F08

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 96
T34 WRITE DATA ADDRESS GREATER THAN 64K

SEQ 0096

```

5079
5080 013606 104421          TCHKOP
5081 013610 104004          ERROR 4 ;OR 5, 6, 7, 10 ;CHECK OPERATION FOR ANY ERRORS
5082 013612 004437 042636 JSR R4,GENCOM ;REPORT ALL ERRORS
5083 013616 002007          2007 ;CLEAR Ibuff TO 1'S
5084 013620 001000          1000
5085
5086 013622 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5087 013626 000121          RDDATA ;RDDATA
5088 013630 177400          -400 ;-400 WORDS
5089 013632 070414          Ibuff ;IBuff IS Buff ADDRESS
5090 013634 020           .BYTE 20 ;SECTOR 20
5091 013635 000           .BYTE 0 ;TRACK 0
5092 013636 000312          312 ;CYLINDER 312
5093 013640 104417          TLOADRK ;LOAD RK REGS
5094 013642 104424          TWAT32 ;WAIT FOR INTERRUPT
5095 013644 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5096
5097 013646 104421          TCHKOP
5098 013650 104004          ERROR 4 ;OR 5, 6, 7, 10 ;CHECK OPERATION FOR ANY ERRORS
5099 013652 004437 042636 JSR R4,GENCOM ;REPORT ALL ERRORS
5100 013656 110000          110000 ;CHECK DATA
5101 013660 003777          3777 ;MEMORY MANAGEMENT WITH DESTINATION
5102 013662 000400          400 ;RELOCATION ARGUMENT
5103 013664 000413          BR 5$ ;400 WORDS
5104 013666 104015          ERROR 15 ;NO MISCOMPARES-SKIP
5105                                     ;REPORT FIRST ERROR
5106 013670 013700 001634 MOV ERRLMT,R0 ;GET ERROR LIMIT
5107 013674 005300          DEC R0 ;DECREMENT COUNT
5108 013676 001406          BEQ 65$ ;IF ZERO - EXIT
5109 013700 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5110 013704 050000          50000
5111 013706 000402          BR 65$ ;NO MORE ERRORS - EXIT
5112 013710 104016          ERROR 16 ;REPORT NEXT ERROR
5113 013712 000770          BR 64$ ;LOOP
5114
5115 013714          65$:
5116          5$:
5117          ;*****
5118          ;*TEST 35 READ DATA ADDRESS GREATER THAN 64K
5119          ;*
5120          ;* ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
5121          ;* CHECK MEMORY FOR CORRECT TRANSFER.
5122          ;*
5123          ;* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
5124          ;* OF MEMORY IS PRESENT.
5125          ;*****
5126 013714 000004          TST35: SCOPE
5127 013716 012737 000012 001262 MOV #10,$TIMES ;DO 10. ITERATIONS
5128 013724 123727 001326 000002 CMPB $MAMS1,#2 ;CHECK IF >64K MEMORY
5129 013732 002001          BGE 2$ ;YES-SKIP
5130 013734 000462          1$: BR 5$ ;EXIT
5131
5132 013736 012737 000032 001110 2$: MOV #32,$LPERR ;SET LOCAL LOOP ON ERROR
5133
5134 013744          3$:

```

```

5135 013744 104416 TSSINIT ;CLEAR SUBSYSTEM
5136 013746 104003 ERROR 3 ;BAD INIT ERROR
5137 013750 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5138 013754 000123 WRDATA ;WRDATA
5139 013756 177400 -400 ;-400 WORDS
5140 013760 072414 OBUFF ;OBUFF IS BUFF ADDRESS
5141 013762 021 .BYTE 21 ;SECTOR 21
5142 013763 000 .BYTE 0 ;TRACK 0
5143 013764 000312 312 ;CYLINDER 312
5144 013766 004437 042636 JSR R4,GENCOM ;GENERATE DATA
5145 013772 000012 12 ;PATTERN 12
5146 013774 000400 400 ;400 WORDS
5147
5148 013776 104417 TLOADRK ;LOAD RK REGS
5149 014000 104430 TWAT96 ;WAIT FOR INTERRUPT
5150 014002 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5151
5152 014004 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5153 014006 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5154 014010 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5155 014014 000521 RDDATA:BA16 ;READ DATA AND SET BA16
5156 014016 177400 -400 ;400 WORDS
5157 014020 177770 177770 ;ACROSS 64K BOUNDARY
5158 014022 021 .BYTE 21 ;FROM SECTOR 21
5159 014023 000 .BYTE 0 ;TRACK 0
5160 014024 000312 312 ;CYLINDER 312
5161
5162 014026 104417 TLOADRK ;LOAD RK REGS
5163 014030 104424 TWAT32 ;WAIT FOR INTERRUPT
5164 014032 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5165
5166 014034 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5167 014036 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5168 014040 004437 042636 JSR R4,GENCOM ;COMPARE DATA
5169 014044 120000 120000 ;MEM MANAGEMENT WITH SOURCE
5170 014046 003777 3777 ;RELOCATION ARGUMENT
5171 014050 000400 400 ;400 WORDS
5172 014052 000413 BR 5$ ;NO MISCOMPARES-SKIP
5173 014054 104015 ERROR 15 ;REPORT FIRST ERROR
5174
5175 014056 013700 001634 MOV ERRLMT,R0 ;GET ERROR LIMIT
5176 014062 005300 64$: DEC R0 ;DECREMENT COUNT
5177 014064 001406 BEQ 65$ ;IF ZERO - EXIT
5178 014066 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5179 014072 060000 60000
5180 014074 000402 BR 65$ ;NO MORE ERRORS - EXIT
5181 014076 104016 ERROR 16 ;REPORT NEXT ERROR
5182 014100 000770 BR 64$ ;LOOP
5183 014102
5184
5185 014102 5$:
5186 ;*****
5187 ;*TEST 36 WRITE DATA ADDRESS GREATER THAN 96K
5188 ;*
5189 ;*
5190 ;* ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
;* MAKE SURE CORRECT DATA IS ON DISK.

```

H08

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 98
T36 WRITE DATA ADDRESS GREATER THAN 96K

SEQ 0098

```

S191
S192
S193
S194
S195
S196 014102 000004
S197 014104 012737 000012 001262
S198 014112 123727 001326 000003
S199 014120 002016
S200 014122 032737 000002 001664 1$:
S201 014130 001011
S202
S203 014132 104401 052135
S204 014136 104401 052310
S205 014142 104401 052314
S206 014146 052737 000002 001664
S207 014154 000463 2$:
S208
S209 014156 012737 014164 001110 3$:
S210
S211 014164 4$:
S212 014164 104416
S213 014166 104003
S214 014170 004437 036252
S215 014174 001123
S216 014176 177400
S217 014200 177770
S218 014202 022
S219 014203 000
S220 014204 000312
S221 014206 004437 042636
S222 014212 010013
S223 014214 005777
S224 014216 000400
S225
S226 014220 104417
S227 014222 104430
S228 014224 104002
S229
S230 014226 104421
S231 014230 104004
S232
S233 014232 004437 036252
S234 014236 000121
S235 014240 177400
S236 014242 070414
S237 014244 022
S238 014245 000
S239 014246 000312
S240
S241 014250 104417
S242 014252 104424
S243 014254 104002
S244
S245 014256 104421
S246 014260 104004

::
:*
:* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
:* OF MEMORY IS PRESENT.
:*
:*****
TST36: SCOPE
MOV #10,$TIMES ;DO 10. ITERATIONS
CMPB $MAMS1,#3 ;CHECK IF >96K MEMORY
BGE 3$ ;YES-SKIP
BIT #MEMSZB,OPTFLG ;TEST IF REPORT FLAG SET
BNE 2$ ;NO-SKIP

TYPE ,OPR011 ;"INSUFFICIENT MEMORY-DATA TRANSFET WITH
TYPE ,OPR014 ;ADDRESS >96K BYPASSED"
TYPE ,OPR015
BIS #MEMSZB,OPTFLG ;SET REPORT FLAG
BR 6$

MOV #4,$SLPERR ;SET LOCAL LOOP ON ERROR

TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REG
WRDATA:BA17 ;WRITE DATA AND BA17
-400 ;400 WORDS FROM
177770 ;ACROSS 96K BOUNDARY
.BYTE 22 ;TO SECTOR 22
.BYTE 0 ;TRACK 0
312 ;CYL 312
JSR R4,GENCOM ;GENERATE DATA
10013 ;PATTERN 13 MEM MAN WITH DEST.
5777 ;RELOCATION ARGUMENT
400 ;400 WORDS

TLOADRK ;LOAD RK REGS
TWTAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

JSR R4,LRLOAD ;LOAD "L" REGS
RDDATA ;RDDATA
-400 ;-400 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 22 ;SECTOR 22
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

TLOADRK ;LOAD RK REGS
TWTAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

```

```

5247
5248 014262 004437 042636 JSR R4,GENCOM ;COMPARE DATA
5249 014266 110000 110000 ;MEM MANAGEMENT WITH DESTINATION
5250 014270 005777 5777 ;RELOCATION ARGUMENT
5251 014272 000400 400 ;400 WORDS
5252 014274 000413 BR 6$ ;NO MISCOMPARES-BRANCH
5253 014276 104015 ERROR 15 ;REPORT 1ST ERROR
5254
5255 014300 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
5256 014304 005300 64$: DEC RO ;DECREMENT COUNT
5257 014306 001406 BEQ 65$ ;IF ZERO - EXIT
5258 014310 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5259 014314 050000 50000
5260 014316 000402 BR 65$ ;NO MORE ERRORS - EXIT
5261 014320 104016 ERROR 16 ;REPORT NEXT ERROR
5262 014322 000770 BR 64$ ;LOOP
5263
5264
5265 014324 65$:
5266 *****
5267 *TEST 37 READ DATA ADDRESS GREATER THAN 96K
5268 *
5269 * ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
5270 * CHECK MEMORY FOR CORRECT TRANSFER.
5271 *
5272 * NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5273 * OF MEMORY IS PRESENT.
5274 *
5275 *****
5276 014324 000004 ST37: SCOPE
5277 014326 012737 000012 001262 MOV #10,$TIMES ;DO 10. ITERATIONS
5278 014334 123727 001326 000003 CMPB $MAMS1,#3 ;CHECK IF >96K MEMORY
5279 014342 002001 BGE 3$ ;YES-SKIP
5280 014344 000462 2$: BR 6$
5281
5282 014346 012737 014354 001110 3$: MOV #4,$SLPERR ;SET LOCAL LOOP ON ERROR
5283
5284 014354 4$:
5285 014354 104416 TSSINIT ;CLEAR SUBSYSTEM
5286 014356 104003 ERROR 3 ;BAD INIT ERROR
5287
5288 014360 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5289 014364 000123 WRDATA ;WRDATA
5290 014366 177400 -400 ;-400 WORDS
5291 014370 072414 OBUFF ;OBUFF IS BUFF ADDRESS
5292 014372 005 .BYTE 5 ;SECTOR 5
5293 014373 000 .BYTE 0 ;TRACK 0
5294 014374 000312 312 ;CYLINDER 312
5295 014376 004437 042636 JSR R4,GENCOM ;GENERATE DATA
5296 014402 000014 14 ;PATTERN 14
5297 014404 000400 400 ;400 WORDS
5298
5299 014406 104417 TLOADRK ;LOAD RK REGS
5300 014410 104430 TWAT96 ;WAIT FOR INTERRUPT
5301 014412 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5302

```

JOB

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 100
T37 READ DATA ADDRESS GREATER THAN 96K

SEQ 0100

```

5303 014414 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5304 014416 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5305 014420 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5306 014424 001121 RDATA!BA17 ;READ DATA WITH BA17 SET
5307 014426 177400 -400 ;400 WORDS
5308 014430 177770 177770 ;ACROSS 96K BOUNDARY
5309 014432 005 .BYTE 5 ;FROM SECTOR 5
5310 014433 000 .BYTE 0 ;TRACK 0
5311 014434 000312 312 ;CYL 312
5312
5313 014436 104417 TLOADRK ;LOAD RK REGS
5314 014440 104424 TWAT32 ;WAIT FOR INTERRUPT
5315 014442 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5316
5317 014444 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5318 014446 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5319 014450 004437 042636 JSR R4,GENCOM ;COMPARE DATA
5320 014454 120000 120000 ;MEM MANAGEMENT WITH SOURCE
5321 014456 005777 5777 ;RELOCATION ARGUMENT
5322 014460 000400 400 ;400 WORDS
5323 014462 000413 BR 6$ ;NO MISCOMPARES-SKIP
5324 014464 104015 ERROR 15 ;REPORT FIRST ERROR
5325
5326 014466 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
5327 014472 005300 64$: DEC RO ;DECREMENT COUNT
5328 014474 001406 BEQ 65$ ;IF ZERO - EXIT
5329 014476 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5330 014502 060000 60000
5331 014504 000402 BR 65$ ;NO MORE ERRORS - EXIT
5332 014506 104016 ERROR 16 ;REPORT NEXT ERROR
5333 014510 000770 BR 64$ ;LOOP
5334 014512
5335
5336 014512 65$:
5337 *****
5338 ;TEST 40 PARTIAL SECTOR WRITE DATA
5339 *****
5340 ;
5341 ; ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,
5342 ; HEAD 0, SECTOR 0. MAKE SURE THE SECTOR WAS
5343 ; ZERO FILLED CORRECTLY.
5344 ;
5345 *****
5346 TST40: SCOPE
5347 014514 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
5348 014522 104416 TSSINIT ;CLEAR SUBSYSTEM
5349 014524 104003 ERROR 3 ;BAD INIT ERROR
5350
5351 014526 004437 036252 JSR R4,LRLOAD ;LOAD "L" REG
5352 014532 000123 WRDATA ;WRITE DATA
5353 014534 177675 -103 ;WORD COUNT PARTIAL SECTOR
5354 014536 072414 OBUFF ;BUFF ADDRESS
5355 014540 007 .BYTE 7 ;SECTOR 7
5356 014541 000 .BYTE 0 ;TRACK 0
5357 014542 000312 312 ;CYLINDER 312
5358 014544 004437 042636 JSR R4,GENCOM ;GENERATE DATA

```

K08

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 101
T40 PARTIAL SECTOR WRITE DATA

SEQ 0101

```

5359 014550 000003          3          ;PATTERN 3
5360 014552 000400          400        ;400 WORDS
5361
5362 014554 104417          TLOADRK    ;LOAD RK REGS
5363 014556 104430          TWAT96     ;WAIT FOR INTERRUPT
5364 014560 104002          ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
5365
5366 014562 104421          TCHKOP     ;CHECK FOR ANY ERROR
5367 014564 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERROR
5368
5369 014566 004437 036252    JSR R4,LRLOAD ;LOAD "L" REGS
5370
5371 014572 000121          RDDATA     ;READ DATA
5372 014574 177400          -400      ;ONE FULL SECTOR
5373 014576 070414          IBUFF     ;BUFF ADDRESS
5374 014600 007          .BYTE 7    ;SECTOR 7
5375 014601 000          .BYTE 0    ;TRACK 0
5376 014602 000312          312      ;CYLINDER 312
5377
5378 014604 004437 042636    JSR R4,GENCOM
5379 014610 002007          2007     ;CLEAR IBUFF TO ALL ONES
5380 014612 000400          400
5381
5382 014614 104417          TLOADRK    ;LOAD RK REGS
5383 014616 104424          TWAT32     ;WAIT FOR INTERRUPT
5384 014620 104002          ERROR 2    ;TO SLOW/NOT COMPLETE
5385
5386 014622 104421          TCHKOP     ;CHECK FOR ANY ERRORS
5387 014624 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
5388
5389 014626 012701 072622    MOV #OBUF+206,R1 ;CLEAR THE LAST 205 WORDS
5390 014632 012700 000275    MOV #275,R0    ;OF THE OUTPUT BUFFER TO ZERO
5391 014636 005021          CLR (R1)+    ;TO VERIFY THE PARTIAL SECTOR
5392 014640 005300          DEC R0      ;WRITE 0 FILLED THE SECTOR
5393 014642 001375          BNE 1$
5394 014644 004437 042636    JSR R4,GENCOM
5395 014650 100000          100000    ;COMPARE OBUF & IBUF.
5396 014652 000400          400      ;ALL 400 WORDS
5397 014654 000413          BR 3$      ;NO ERRORS-EXIT
5398 014656 104015          ERROR 15   ;REPORT FIRST COMPARE ERROR
5399
5400 014660 013700 001634    MOV ERLMT,R0 ;GET ERROR LIMIT
5401 014664 005300          DEC R0     ;DECREMENT IT
5402 014666 001406          BEQ 3$    ;IF ZERO-EXIT
5403 014670 004437 042636    JSR R4,GENCOM
5404 014674 040000          40000    ;CONTINUE COMPARE
5405 014676 000402          BR 3$    ;NO MORE ERRORS-EXIT
5406 014700 104016          ERROR 16  ;REPORT NEXT COMPARE ERROR
5407 014702 000770          BR 2$    ;LOOP
5408
5409 014704          3$:
5410          ;*****
5411          ;*TEST 41 PARTIAL SECTOR READ DATA
5412          ;*
5413          ;* WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A
5414          ;* KNOWN CONFIGURATION. ISSUE A READ DATA OF

```

103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0.
MAKE SURE ONLY 103 WORDS GET TRANSFERRED
TO MEMORY.

```

†ST41: SCOPE
MOV #10.,$TIMES ;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

JSR R4,LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-400 ;-400 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312
JSR R4,GENCOM ;GENERATE DATA
4 ;PATTERN 4
400 ;400 WORDS

TLOADRK ;LOAD RK REGS
TWAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

JSR R4,LRLOAD ;LOAD "L" REGS
RDDATA ;RDDATA
-103 ;-103 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312
JSR R4,GENCOM
2007 ;CLEAR IBUFF
400

TLOADRK ;LOAD RK REGS
TWAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

MOV #OBUFF+206,R0 ;AFTER THE LAST 205 WORDS OF
MOV #275,R1 ;THE OUTPUT BUFFER TO ALL ONES.
MOV #-1,(R0)+ ;THESE SHOULD ALL BE ONES IN
DEC R1 ;IBUFF BECAUSE THE PARTIAL
BNE 1$ ;READ FILLED ONLY 103 WORDS.
JSR R4,GENCOM ;GO COMPARE IBUFF & OBUFF
100000
400 ;ALL 400 WORDS
BR 3$ ;NO ERRORS-EXIT
ERROR 15 ;REPORT FIRST COMPARE ERROR

```

```

5415
5416
5417
5418
5419
5420 014704 000004
5421 014706 012737 000012 001262
5422 014714 104416
5423 014716 104003
5424
5425 014720 004437 036252
5426 014724 000123
5427 014726 177400
5428 014730 072414
5429 014732 017
5430 014733 000
5431 014734 000312
5432 014736 004437 042636
5433 014742 000004
5434 014744 000400
5435
5436 014746 104417
5437 014750 104430
5438 014752 104002
5439
5440 014754 104421
5441 014756 104004
5442
5443 014760 004437 036252
5444 014764 000121
5445 014766 177675
5446 014770 070414
5447 014772 017
5448 014773 000
5449 014774 000312
5450 014776 004437 042636
5451 015002 002007
5452 015004 000400
5453
5454 015006 104417
5455 015010 104424
5456 015012 104002
5457 015014 104421
5458 015016 104004
5459
5460 015020 012700 072622
5461 015024 012701 000275
5462 015030 012720 177777
5463 015034 005301
5464 015036 001374
5465 015040 004437 042636
5466 015044 100000
5467 015046 000400
5468 015050 000413
5469 015052 104015
5470

```

1\$:


```

5471 015054 013700 001634      MOV      ERRMT,RO      ;GET ERROR LIMIT
5472 015060 005300      64$: DEC      RO      ;DECREMENT COUNT
5473 015062 001406      BEQ      65$          ;IF ZERO - EXIT
5474 015064 004437 042636      JSR      R4,GENCOM    ;CONTINUE DATA COMPARE
5475 015070 040000      40000
5476 015072 000402      BR       65$          ;NO MORE ERRORS - EXIT
5477 015074 104016      ERROR    16          ;REPORT NEXT ERROR
5478 015076 000770      BR       64$          ;LOOP
5479 015100
5480
5481 015100      3$:
5482 *****
5483 *TEST 42          WRITE DATA WITH NON-EXISTENT MEMORY
5484 *
5485 *          ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
5486 *          MAKE SURE NON-EXISTENT MEMORY SETS.
5487 *
5488 *****
5489 †ST42:          SCOPE
5490 015102 012737 000012 001262      MOV      #10.,$TIMES  ;DO 10. ITERATIONS
5491 015110 104416      TSSINIT          ;CLEAR SUBSYSTEM
5492 015112 104003      ERROR    3        ;BAD INIT ERROR
5493
5494 015114 004437 036252      JSR      R4,LRLOAD   ;LOAD "L" REG
5495 015120 001523      BA16!BA17!WRDATA  ;BA16 & 17 SET WITH WRITE DATA
5496 015122 177777      -1             ;WORD COUNT OF 1
5497 015124 176000      176000        ;BUFF ADDRESS=10 PAGE BASE
5498 015126 013          .BYTE    13      ;SECT 13
5499 015127 000          .BYTE    0        ;TRACK 0
5500 015130 000312      312           ;CYLINDER 312
5501
5502 015132 104417      TLOADRK        ;LOAD RK REGS
5503 015134 104430      TWAT96        ;WAIT FOR INTERRUPT
5504 015136 104002      ERROR    2        ;TO SLOW/NOT COMPLETE ERROR
5505 015140 104422      TCHKWE        ;CHECK OPERATION WITH ERROR
5506 015142 000000      0
5507 015144 000000      0
5508 015146 000040      NEMERR        ;NON-EXISTENT MEMORY ERROR
5509 015150 104004      ERROR    4 :OR5,6,7 ;REPORT ANY DISCREPENCIES
5510 015152 012737 054513 001450      MOV      #EM11A,EM10N ;SET UP ERROR MESSAGE
5511 015160 012737 051136 061434      MOV      #OPER42,DF010A ;WITH SUPPORT MESSAGE
5512 015166 113700 001541      MOV      T.CS1+1,RO  ;GET UPPER CS1
5513 015172 042700 177774      BIC      #177774,RO  ;CLEAR UNUSED BITS
5514 015176 022700 000003      CMP      #3,RO      ;TEST IF BOTH UPPER BUS BITS SET
5515 015202 001406      BEQ      1$        ;YES - SKIP
5516 015204 010037 001204      MOV      RO,$REG11  ;SET UP FOR ERROR REPORT
5517 015210 012737 000003 001202      MOV      #3,$REG10
5518 015216 104010      ERROR    10
5519 015220 022737 176002 001544 1$: CMP      #176002,T.BA ;TEST IF BUSS ADDRESS LOW OKAY
5520 015226 001412      BEQ      2$        ;YES - SKIP
5521 015230 012737 054465 001450      MOV      #EM11,EM10N ;SET UP MESSGAE
5522 015236 012737 176002 001202      MOV      #176002,$REG10 ;STORE VALUE FOR REPORT
5523 015244 013737 001544 001204      MOV      T.BA,$REG11
5524 015252 104010      ERROR    10
5525
5526 015254 005737 001542      2$: TST      T.WC      ;TEST IF WORD COUNT CORRECT

```

```

5527 015260 001411          BEQ      3$          ;YES - SKIP
5528 015262 012737 054440 001450  MOV     #EM10,EM1ON ;SET UP MESSAGE
5529 015270 005037 001202          CLR     $REG10
5530 015274 013737 001542 001204  MOV     T.WC,$REG11
5531 015302 104010          ERROR  10
5532
5533 015304          3$:
5534
5535          ;*****
5536          ;TEST 43          READ DATA WITH NON-EXISTENT MEMORY
5537          ;
5538          ;          ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000.
5539          ;          MAKE SURE NON-EXISTENT MEMORY SETS.
5540          ;
5541          ;*****
5542 015304 000004          TST43: SCOPE
5543 015306 012737 000012 001262  MOV     #10.,$TIMES ;DO 10. ITERATIONS
5544 015314 104416          TSSINIT ;CLEAR SUBSYSTEM
5545 015316 104003          ERROR   3          ;BAD INIT ERROR
5546
5547 015320 004437 036252          JSR     R4,LRLOAD ;LOAD "L" REG
5548 015324 001521          BA16!BA17!RDDATA ;BA16 & 17 WITH READ DATA
5549 015326 177777          -1          ;WORD COUNT OF 1
5550 015330 176000          176000 ;BUFF ADDRESS=10 PAGE BASE
5551 015332 013          .BYTE  13 ;SECTOR 13
5552 015333 000          .BYTE  0 ;TRACK 0
5553 015334 000312          312 ;CYL 312
5554
5555 015336 104417          TLOADRK ;LOAD RK REGS
5556 015340 104430          TWAT96 ;WAIT FOR INTERRUPT
5557 015342 104002          ERROR   2          ;TO SLOW/NOT COMPLETE ERROR
5558 015344 104422          TCHKWE ;CHECK OPERATION WITH ERRORS
5559 015346 000000          0
5560 015350 000000          0
5561 015352 000040          NEMERR ;NON-EXISTENT MEMORY ERROR
5562 015354 104004          ERROR   4: OR 5,6,7 ;REPORT ALL DISCREPANCIES
5563 015356 012737 054513 001450  MOV     #EM11A,EM1ON ;SET MESSAGE
5564 015364 012737 051212 061434  MOV     #OPER43,DF010A ;SET SUPPORT MESSAGE
5565 015372 113700 001541          MOV     T.CS1+1,R0 ;GET UPPER CS1
5566 015376 042700 177774          BIC     #177774,R0 ;CLEAR UNWANTED BITS
5567 015402 022700 000003          CMP     #3,R0 ;TEST BOTH BUS 16 & 17 SET
5568 015406 001406          BEQ     1$          ;YES - SKIP
5569 015410 012737 000003 001202  MOV     #3,$REG10 ;SET VALUES FOR REPORT
5570 015416 010037 001204          MOV     R0,$REG11
5571 015422 104010          ERROR  10
5572
5573 015424 022737 176002 001544 1$:  CMP     #176002,T.BA ;TEST IF BUS ADDRESS CORRECT
5574 015432 001412          BEQ     2$          ;YES - SKIP
5575 015434 012737 054465 001450  MOV     #EM11,EM1ON ;SET MESSAGE
5576 015442 012737 176002 001202  MOV     #176002,$REG10 ;SET VALUES FOR REPORT
5577 015450 013737 001544 001204  MOV     T.BA,$REG11
5578 015456 104010          ERROR  10
5579
5580 015460 005737 001542          2$:  TST     T.WC ;TEST IF WORD COUNT CORRECT
5581 015464 001411          BEQ     3$          ;YES - SKIP
5582 015466 012737 054440 001450  MOV     #EM10,EM1ON ;SET MESSAGE

```

```

5583 015474 005037 001202 CLR $REG10 ;SET VALUES
5584 015500 013737 001542 001204 MOV T.WC,$REG11
5585 015506 104010 ERROR 10
5586
5587 015510 3$:
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603 015510 000004
5604 015512 012737 000012 001262
5605 015520 104416
5606 015522 104003
5607
5608 015524 032737 000100 001664
5609 015532 001013
5610 015534 032737 000004 001664
5611 015542 001005
5612 015544 104401 051740
5613 015550 052737 000004 001664
5614 015556 000137 016350 25$:
5615
5616 015562 1$:
5617 015562 004437 036252
5618 015566 000123
5619 015570 177400
5620 015572 072414
5621 015574 010
5622 015575 000
5623 015576 000312
5624
5625 015600 032737 002000 001664
5626 015606 001002
5627 015610 005077 164056
5628
5629 015614 004437 042636 3$:
5630 015620 000007
5631 015622 000400
5632
5633 015624 012746 000340
5634 015630 012746 015636
5635 015634 000002
5636 015636 10$:
5637
5638 015636 012737 100200 072640 MOV #100200,OBUFF+224 ;SET WORD IN BUFFER

```

```

*****
*TEST 44 UNIBUS PARITY ERROR
*****
*
* INITIALIZE A MEMORY LOCATION WITH BAD
* PARITY. ISSUE A WRITE DATA OF 400 WORDS
* STARTING AT A LOCATION 112 WORDS BEFORE
* THE LOCATION WITH BAD PARITY. MAKE SURE
* THAT UNIBUS PARITY ERROR SETS.
*
* NOTE: THIS TEST IS ONLY EXECUTED IF
* MEMORY PARITY OPTION EXISTS FOR
* BUFFER.
*****

```

```

*****
*ST44: SCOPE
MOV #10.,$TIMES ;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
*
* BIT #PARPRE,OPTFLG ;TEST IF PARITY OPTION PRESENT
* BNE 1$ ;YES-SKIP
* BIT #MEMPYB,OPTFLG ;TEST IF PARITY OPTION REPORTED
* BNE 25$ ;YES-SKIP TO EXIT
* TYPE OPRO10 ;PRINT BYPASS MESSAGE
* BIS #MEMPYB,OPTFLG ;SET OPTION REPORTED BIT
* JMP 4$ ;SKIP TO EXIT
*
* JSR R4,LRLOAD ;LOAD "L" REGS
* WRDATA ;WRDATA
* -400 ;-400 WORDS
* OBUFF ;OBUFF IS BUFF ADDRESS
* .BYTE 10 ;SECTOR 10
* .BYTE 0 ;TRACK 0
* 312 ;CYLINDER 312
*
* BIT #CP1170,OPTFLG ;TEST IF 11/70
* BNE 3$ ;YES - SKIP
* CLR @CSRPTR ;CLEAR PARITY IE
*
* JSR R4,GENCOM ;GENERATE DATA
* 7 ;PATTERN 7
* 400 ;400 WORDS
*
* MOV #PR7,-(SP) ;PUT PRIORITY 7 ON STACK
* MOV #10$,-(SP) ;PUT ADDRESS ON STACK
* RTI ;SET PRI

```

```

5639 015644 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5640 015652 001011          BNE      55$        ;YES - SKIP
5641 015654 012777 000004 164010      MOV      #BIT2,@CSRPTR ;SET WRITE WRONG PARITY BIT
5642 015662 012737 100200 072640      MOV      #100200,OBUFF+224 ;SET BAD PARITY IN MEMORY
5643 015670 012777 000001 163774      MOV      #BIT0,@CSRPTR ;CLEAR CONTROL BIT, SET IE BIT
5644
5645 015676 013746 001622          55$:      MOV      RKPRI,-(SP) ;SET OLD PRIORITY
5646 015702 012746 015710          MOV      #11$,-(SP) ;SET ADDRESS
5647 015706 000002          RTI      ;RESTORE PRI
5648 015710 013704 001672          11$:      MOV      CSRPTR,R4 ;SET R4 WITH CSR POINTER
5649 015714 005000          CLR      R0 ;SET R0 FOR COUNTER
5650 015716 012701 001662          MOV      #INTSET,R1 ;SET R1 FOR POINTER TO INTERRUPT FLAG
5651 015722 012777 070400 163670      MOV      #SPCHLR,@RKVEC ;SET UP INTERRUPT VECTORS FOR
5652 015730 012777 070410 163760      MOV      #SPCPAR,@MMVECA ;RK611 AND PARITY ERROR
5653 015736 012737 016324 001264      MOV      #2$, $ESCAPE ;SET UP ESCAPE FOR ERROR
5654 015744 104417          TLOADRK ;LOAD RK REGS
5655 015746 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5656 015754 001434          BEQ      45$        ;NO - SKIP
5657 015756 012737 000016 177746      MOV      #16,177746 ;SET TO DISABLE CACHE AND UNIBUS ERROR
5658 015764 012777 000000 163716      MOV      #0,@KWLADD ;TURN OFF CLOCK INTERRUPTS
5659 015772 012777 000000 163144      MOV      #0,@STKS ;TURN OFF KEYBOARD
5660 016000 012737 170000 177750      MOV      #170000,177750 ;SET FOR ERROR FORCE
5661
5662          ;*****
5663 016006 105711          40$:      TSTB      (R1) ;LOOP TO WAIT FOR INTERRUPT OR ABORT
5664 016010 003005          BGT      43$        ;WAIT. THE CODE BETWEEN THE STARS IS SET
5665 016012 005300          DEC      R0 ;SET UP SO ALL BYTES HAVE PARITY OF 1.
5666 016014 100774          BMI      40$        ;IF THIS CODE IS CHANGED, REMEMBER ALL
5667 016016 000240          NOP ;BYTES MUST HAVE AN EVEN NUMBER OF
5668 016020 003372          BGT      40$        ;BITS.
5669 016022 000240          NOP
5670 016024 005014          43$:      CLR      (R4) ;CLEAR ERROR FORCE
5671          ;*****
5672
5673 016026 005037 177746          CLR      177746 ;ENABLE CACHE
5674 016032 012777 000100 163650      MOV      #BIT6,@KWLADD ;TURN ON CLOCK INTERRUPTS
5675 016040 012777 000100 163076      MOV      #100,@STKS ;TURN ON KEYBOARD INTERRUPTS
5676 016046 104430          45$:      TWAT96 ;WAIT FOR INTERRUPT
5677 016050 000414          BR      46$        ;TO SLOW/NOT COMPLETE ERROR
5678 016052 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5679 016060 001024          BNE      48$        ;YES - SKIP
5680 016062 005077 163604          CLR      @CSRPTR ;ELSE CLEAR CSR
5681 016066 005037 072640          CLR      OBUFF+224 ;CLEAR THE BAD PARITY WORD
5682 016072 012777 000001 163572      MOV      #1,@CSRPTR ;SET PARITY DETECT AGAIN
5683 016100 000414          BR      48$        ;SKIP
5684
5685 C16102 032737 002000 001664 46$:      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5686 C16110 001007          BNE      47$        ;YES - SKIP
5687 016112 005077 163554          CLR      @CSRPTR ;CLEAR CSR
5688 016116 005037 072640          CLR      OBUFF+224 ;CLEAR BAD PARITY WORD
5689 016122 012777 000001 163542      MOV      #1,@CSRPTR ;SET UP PARITY DETECT AGAIN
5690 016130 104002          47$:      ERROR 2 ;REPORT TO SLOW ERROR
5691 016132          48$:
5692 016132 104422          TCHKWE ;CHECK OPERATION WITH ERROR
5693 016134 000000          0
5694 016136 000000          0

```

```

5695 016140 000100 UPERR ;UNIBUS PARITY ERROR
5696 016142 104004 ERROR 4; OR 5,6,7 ;REPORT ALL DISCREPANCIES
5697
5698 016144 005037 001264 CLR $ESCAPE ;CLEAR ESCAPE
5699 016150 012737 051265 061434 MOV #OPER44,DF010A ;SET MESSAGES FOR REPORT
5700 016156 012737 054465 001450 MOV #EM11,EM10N
5701 016164 023727 001544 072642 CMP T.BA,#OBUFF+226 ;CHECK IF BA IN RANGE
5702 016172 103010 BHIS 14$ ;NOT TO LOW - SKIP
5703 016174 012737 072642 001202 MOV #OBUFF+226,$REG10 ;SET VALUES FOR REPORT
5704 016202 013737 001544 001204 MOV T.BA,$REG11
5705 016210 104010 ERROR 10
5706 016212 000413 BR 16$
5707
5708 016214 023727 001544 072646 14$: CMP T.BA,#OBUFF+232 ;CHECK IF BA IN RANGE
5709 016222 101407 BLOS 16$ ;YES - SKIP
5710 016224 012737 072646 001202 MOV #OBUFF+232,$REG10 ;SET VALUES
5711 016232 012737 001544 001204 MOV #T.BA,$REG11
5712 016240 104010 ERROR 10
5713
5714 016242 012737 054440 001450 16$: MOV #EM10,EM10N ;SET MESSAGE
5715 016250 023727 001542 177513 CMP T.WC,#-265 ;CHECK IF WORD COUNT WITHIN RANGE
5716 016256 103007 BHIS 20$ ;YES - SKIP
5717 016260 012737 177513 001202 MOV #-265,$REG10 ;SET VALUES
5718 016266 013737 001542 001204 MOV T.WC,$REG11
5719 016274 104010 ERROR 10
5720
5721 016276 023727 001542 177515 20$: CMP T.WC,#-263 ;STILL CHECKING IF WC IN RANGE
5722 016304 101407 BLOS 2$ ;YES - SKIP
5723 016306 012737 177515 001202 MOV #-263,$REG10 ;SET VALUES
5724 016314 013737 001542 001204 MOV T.WC,$REG11
5725 016322 104010 ERROR 10
5726
5727 016324 2$:
5728 016324 005037 001264 CLR $ESCAPE ;REV 006
5729 016330 012777 035040 163262 MOV #INTHLR,ARKVEC ;RESET INT. VECTORS FOR RK06
5730 016336 004737 035120 JSR PC,OPTTST
5731 016342 012777 000100 162574 MOV #100,$STKS ;INSURE KEYBOARD IS ENABLED
5732
5733 016350 4$:
5734
5735
5736 .SBTTL **MULTIPLE SECTOR OPERATIONS
5737
5738 ;*****
5739 ;*TEST 45 TWO SECTOR WRITE DATA (PART 1)
5740 ;*
5741 ;* ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5742 ;* TRACK 0, SECTOR 0. READ DATA BACK ONE SECTOR
5743 ;* AT A TIME AND MAKE SURE IT IS CORRECT.
5744 ;*
5745 ;*****
5746 †ST45: SCOPE
5747 016352 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
5748 016360 104416 TSSINIT ;CLEAR SUBSYSTEM
5749 016362 104003 ERROR 3 ;BAD INIT ERROR
5750

```

5751	016364	004437	036252	JSR	R4,LRLOAD	;LOAD "L" REGS
5752	016370	000123		WRDATA		;WRDATA
5753	016372	177000		-1000		; -1000 WORDS
5754	016374	072414		OBUFF		; OBUFF IS BUFF ADDRESS
5755	016376	000		.BYTE	0	; SECTOR 0
5756	016377	000		.BYTE	0	; TRACK 0
5757	016400	000312		312		; CYLINDER 312
5758						
5759	016402	004437	042636	JSR	R4,GENCOM	;GENERATE DATA
5760	016406	000015		15		;PATTERN 15
5761	016410	001000		1000		;1000 WORDS
5762						
5763	016412	104417		TLOADRK		;LOAD RK REGS
5764	016414	104430		TWAT96		;WAIT FOR INTERRUPT
5765	016416	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5766						
5767	016420	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5768	016422	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5769						
5770	016424	004437	042636	JSR	R4,GENCOM	;CLEAR IBUFF
5771	016430	002007		2007		;TO ALL 1'S
5772	016432	001000		1000		
5773						
5774	016434	004437	036252	JSR	R4,LRLOAD	;LOAD "L" REGS
5775	016440	000121		RDDATA		;RDDATA
5776	016442	177400		-400		; -400 WORDS
5777	016444	070414		IBUFF		; IBUFF IS BUFF ADDRESS
5778	016446	000		.BYTE	0	; SECTOR 0
5779	016447	000		.BYTE	0	; TRACK 0
5780	016450	000312		312		; CYLINDER 312
5781						
5782	016452	104417		TLOADRK		;LOAD RK REGS
5783	016454	104424		TWAT32		;WAIT FOR INTERRUPT
5784	016456	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5785						
5786	016460	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5787	016462	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5788						
5789	016464	004437	036252	JSR	R4,LRLOAD	;LOAD "L" REGS
5790	016470	000121		RDDATA		;RDDATA
5791	016472	177400		-400		; -400 WORDS
5792	016474	071414		IBUFF+1000		; IBUFF+1000 IS BUFF ADDRESS
5793	016476	001		.BYTE	1	; SECTOR 1
5794	016477	000		.BYTE	0	; TRACK 0
5795	016500	000312		312		; CYLINDER 312
5796						
5797	016502	104417		TLOADRK		;LOAD RK REGS
5798	016504	104424		TWAT32		;WAIT FOR INTERRUPT
5799	016506	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5800						
5801	016510	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5802	016512	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5803						
5804	016514	004437	042636	JSR	R4,GENCOM	;COMPARE DATA
5805	016520	100000		100000		
5806	016522	001000		1000		;1000 WORDS

```

5807 016524 000413 BR 25 ;NO MISCOMPARES-EXIT
5808 016526 104015 ERROR 15 ;REPORT FIRST ERROR
5809
5810 016530 013700 001634 MOV ERRLMT,R0 ;GET ERROR LIMIT
5811 016534 005300 64$: DEC R0 ;DECREMENT COUNT
5812 016536 001406 BEQ 65$ ;IF ZERO - EXIT
5813 016540 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5814 016544 040000 40000
5815 016546 000402 BR 65$ ;NO MORE ERRORS - EXIT
5816 016550 104016 ERROR 16 ;REPORT NEXT ERROR
5817 016552 000770 BR 64$ ;LOOP
5818 016554 65$:
5819
5820 016554 2$:
5821 *****
5822 ;TEST 46 TWO SECTOR WRITE DATA (PART 2)
5823 ;
5824 ; ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5825 ; TRACK 0, SECTOR 23. READ DATA BACK ONE SECTOR
5826 ; AT A TIME AND MAKE SURE A MID-TRANSFER
5827 ; SEEK DID NOT TAKE PLACE.
5828 ;
5829 *****
5830 016554 000004 ST46: SCOPE
5831 016556 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
5832 016564 104416 TSSINIT ;CLEAR SUBSYSTEM
5833 016566 104003 ERROR 3 ;BAD INIT ERROR
5834
5835 016570 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5836 016574 000123 WRDATA ;WRDATA
5837 016576 177000 -1000 ;-1000 WORDS
5838 016600 072414 OBUFF ;OBUFF IS BUFF ADDRESS
5839 016602 023 .BYTE 23 ;SECTOR 23
5840 016603 000 .BYTE 0 ;TRACK 0
5841 016604 000312 312 ;CYLINDER 312
5842
5843 016606 004437 042636 JSR R4,GENCOM ;GENERATE DATA
5844 016612 000016 16 ;PATTERN 16
5845 016614 001000 1000 ;1000 WORDS
5846
5847 016616 104417 TLOADRK ;LOAD RK REGS
5848 016620 104430 TWAT96 ;WAIT FOR INTERRUPT
5849 016622 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5850
5851 016624 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5852 016626 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5853
5854 ; IF THE TRACK ADDRESS AT THE END OF OPERATION IS IN ERROR
5855 ; THE CONTROLLER DID A MID-TRANSFER SEEK AS THOUGH IT
5856 ; WERE IN 24(8) SECTORS PER TRACK MODE.
5857 ;
5858 016630 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
5859 016634 000121 RDDATA ;RDDATA
5860 016636 177400 -400 ;-400 WORDS
5861 016640 070414 IBUFF ;IBUFF IS BUFF ADDRESS
5862 016642 023 .BYTE 23 ;SECTOR 23

```

```

5863 016643 000          .BYTE 0          ;TRACK 0
5864 016644 000312      312          ;CYLINDER 312
5865
5866 016646 004437 042636 JSR      R4,GENCOM ;CLEAR IBUFF TO ALL ONES
5867 016652 002007      2007
5868 016654 001000      1000
5869
5870 016656 104417      TLOADRK ;LOAD RK REGS
5871 016660 104424      TWAT32 ;WAIT FOR INTERRUPT
5872 016662 104002      ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5873
5874 016664 104421      TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5875 016666 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5876
5877 016670 004437 036252 JSR      R4,LRLoad ;LOAD "L" REGS
5878 016674 000121      RDDATA ;RDDATA
5879 016676 177400      -400 ;-400 WORDS
5880 016700 071414      Ibuff+1000 ;IBUFF+1000 IS BUFF ADDRESS
5881 016702 024          .BYTE 24 ;SECTOR 24
5882 016703 000          .BYTE 0 ;TRACK 0
5883 016704 000312      312 ;CYLINDER 312
5884
5885 016706 104417      TLOADRK ;LOAD RK REGS
5886 016710 104424      TWAT32 ;WAIT FOR INTERRUPT
5887 016712 104002      ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5888
5889 016714 104421      TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5890 016716 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5891
5892 016720 004437 042636 JSR      R4,GENCOM ;COMPARE DATA
5893 016724 100000      100000
5894 016726 001000      1000 ;1000 WORDS
5895 016730 000413      BR      15 ;NO ERRORS-SKIP
5896 016732 104015      ERROR 15 ;REPORT FIRST ERROR
5897
5898 016734 013700 001634 MOV      ERRlMT,R0 ;GET ERROR LIMIT
5899 016740 005300      DEC      R0 ;DECREMENT COUNT
5900 016742 001406      BEQ     65$ ;IF ZERO - EXIT
5901 016744 004437 042636 JSR      R4,GENCOM ;CONTINUE DATA COMPARE
5902 016750 040000      40000
5903 016752 000402      BR      65$ ;NO MORE ERRORS - EXIT
5904 016754 104016      ERROR 16 ;REPORT NEXT ERROR
5905 016756 000770      BR      64$ ;LOOP
5906 016760
5907 016760
5908
5909
5910
5911
5912
5913
5914
5915
5916 016760 000004      ST47: SCOPE
5917 016762 012737 000012 001262 MOV      #10.,$TIMES ;DO 10. ITERATIONS
5918 016770 104416      TSSINIT ;CLEAR SUBSYSTEM

```

64\$:

65\$:

1\$:

*TEST 47 TWO SECTOR WRITE DATA (PART 3)

*
* ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312,
* TRACK 0, SECTOR 10. READ DATA BACK ONE SECTOR AT
* A TIME AND CHECK ZERO FILL OF SECOND SECTOR.
*

5919	016772	104003		ERROR	3		;BAD INIT ERROR
5920							
5921	016774	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
5922	017000	000123		WRDATA			;WRDATA
5923	017002	177377		-401			; -401 WORDS
5924	017004	072414		OBUFF			; OBUFF IS BUFF ADDRESS
5925	017006	010		.BYTE	10		;SECTOR 10
5926	017007	000		.BYTE	0		;TRACK 0
5927	017010	000312		312			;CYLINDER 312
5928							
5929	017012	004437	042636	JSR	R4,GENCOM		;GENERATE DATA
5930	017016	000002		2			;PATTERN 2
5931	017020	001000		1000			;1000 WORDS
5932							
5933	017022	104417		TLOADRK			;LOAD RK REGS
5934	017024	104430		TWAT96			;WAIT FOR INTERRUPT
5935	017026	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
5936							
5937	017030	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
5938	017032	104004		ERROR	4	;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
5939				CLEAR	LAST 377 WORDS OF		OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
5940	017034	012700	073416	MOV	#OBUFF+1002,R0		;GET STARTING ADDRESS TO BE CLEARED
5941	017040	012701	000377	MOV	#377,R1		;NUMBER OF WORDS
5942	017044	005020		CLR	(R0)+		;CLEAR WORD
5943	017046	005301		DEC	R1		;DEC COUNT
5944	017050	001375		BNE	1\$;LOOP UNTIL COUNT ZERO
5945	017052	004437	042636	JSR	R4,GENCOM		;CLEAR IBUFF TO ONES
5946	017056	002007		2007			
5947	017060	001000		1000			
5948							
5949	017062	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
5950	017066	000121		RDDATA			;RDDATA
5951	017070	177400		-400			; -400 WORDS
5952	017072	070414		IBUFF			;IBUFF IS BUFF ADDRESS
5953	017074	010		.BYTE	10		;SECTOR 10
5954	017075	000		.BYTE	0		;TRACK 0
5955	017076	000312		312			;CYLINDER 312
5956							
5957	017100	104417		TLOADRK			;LOAD RK REGS
5958	017102	104424		TWAT32			;WAIT FOR INTERRUPT
5959	017104	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
5960							
5961	017106	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
5962	017110	104004		ERROR	4	;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
5963	017112	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
5964	017116	000121		RDDATA			;RDDATA
5965	017120	177400		-400			; -400 WORDS
5966	017122	071414		IBUFF+1000			;IBUFF+1000 IS BUFF ADDRESS
5967	017124	011		.BYTE	11		;SECTOR 11
5968	017125	000		.BYTE	0		;TRACK 0
5969	017126	000312		312			;CYLINDER 312
5970							
5971	017130	104417		TLOADRK			;LOAD RK REGS
5972	017132	104424		TWAT32			;WAIT FOR INTERRUPT
5973	017134	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
5974							

```

5975 017136 004437 042636 JSR R4,GENCOM ;DATA COMPARE
5976 017142 100000 100000 ;1000 WORDS
5977 017144 001000 1000 ;NO ERROR-SKIP
5978 017146 000413 BR 2$ ;REPORT FIRST ERROR
5979 017150 104015 ERROR 15
5980
5981 017152 013700 001634 MOV ERRHLT,RO ;GET ERROR LIMIT
5982 017156 005300 64$: DEC RO ;DECREMENT COUNT
5983 017160 001406 BEQ 65$ ;IF ZERO - EXIT
5984 017162 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5985 017166 040000 40000
5986 017170 000402 BR 65$ ;NO MORE ERRORS - EXIT
5987 017172 104016 ERROR 16 ;REPORT NEXT ERROR
5988 017174 000770 BR 64$ ;LOOP
5989 017176 65$:
5990
5991 017176 2$:
5992 *****
5993 *TEST 50 MID-TRANSFER SEEK ON WRITE (PART 1)
5994 *
5995 * ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5996 * TRACK 0, SECTOR 25. READ DATA BACK ONE SECTOR
5997 * AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5998 * DID TAKE PLACE.
5999 *
6000 *****
6001 017176 000004 TST50: SCOPE
6002 017200 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
6003 017206 104416 TSSINIT ;CLEAR SUBSYSTEM
6004 017210 104003 ERROR 3 ;BAD INIT ERROR
6005
6006 017212 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
6007 017216 000123 WRDATA ;WRDATA
6008 017220 177000 -1000 ;-1000 WORDS
6009 017222 072414 OBUFF ;OBUFF IS BUFF ADDRESS
6010 017224 025 .BYTE 25 ;SECTOR 25
6011 017225 000 .BYTE 0 ;TRACK 0
6012 017226 000312 312 ;CYLINDER 312
6013
6014 017230 004437 042636 JSR R4,GENCOM ;GENERATE DATA
6015 017234 000003 3 ;PATTERN 3
6016 017236 001000 1000 ;1000 WORDS
6017
6018 017240 104417 TLOADRK ;LOAD RK REGS
6019 017242 104430 TWAT96 ;WAIT FOR INTERRUPT
6020 017244 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6021
6022 017246 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6023 017250 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6024 ; A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A
6025 ; MID-TRANSFER HEAD SWITCH DID NOT OCCUR.
6026 017252 004437 042636 JSR R4,GENCOM
6027 017256 002007 2007
6028 017260 001000 1000
6029
6030 017262 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS

```

```

6031 017266 000121          RDDATA          ;RDDATA
6032 017270 177400          -400          ; -400 WORDS
6033 017272 070414          Ibuff         ;IBUFF IS BUFF ADDRESS
6034 017274         025      .BYTE 25       ;SECTOR 25
6035 017275         000      .BYTE 0         ;TRACK 0
6036 017276 000312          312          ;CYLINDER 312
6037
6038 017300 104417          TLOADRK       ;LOAD RK REGS
6039 017302 104425          TWAT48        ;WAIT FOR INTERRUPT
6040 017304 104002          ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
6041
6042 017306 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
6043 017310 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6044
6045 017312 004437 036252    JSR R4,LRLOAD ;LOAD "L" REGS
6046 017316 000121          RDDATA          ;RDDATA
6047 017320 177400          -400          ; -400 WORDS
6048 017322 071414          Ibuff+1000    ;IBUFF+1000 IS BUFF ADDRESS
6049 017324         000      .BYTE 0         ;SECTOR 0
6050 017325         001      .BYTE 1         ;TRACK 1
6051 017326 000312          312          ;CYLINDER 312
6052
6053 017330 104417          TLOADRK       ;LOAD RK REGS
6054 017332 104425          TWAT48        ;WAIT FOR INTERRUPT
6055 017334 104002          ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
6056
6057 017336 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
6058 017340 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6059
6060 017342 004437 042636    JSR R4,GENCOM ;COMPARE DATA
6061 017346 100000          100000       ;
6062 017350 001000          1000         ;1000 WORDS
6063 017352 000413          BR 1$        ;NO ERRORS-SKIP
6064 017354 104015          ERROR 15     ;REPORT FIRST ERROR
6065
6066 017356 013700 001634    MOV ERRLMT,RO ;GET ERROR LIMIT
6067 017362 005300          DEC RO        ;DECREMENT COUNT
6068 017364 001406          BEQ 65$      ;IF ZERO - EXIT
6069 017366 004437 042636    JSR R4,GENCOM ;CONTINUE DATA COMPARE
6070 017372 040000          40000       ;
6071 017374 000402          BR 65$       ;NO MORE ERRORS - EXIT
6072 017376 104016          ERROR 16     ;REPORT NEXT ERROR
6073 017400 000770          BR 64$       ;LOOP
6074 017402
6075
6076 017402          1$:
6077          ;*****
6078          ;*TEST 51      MID-TRANSFER SEEK ON WRITE (PART 2)
6079          ;*
6080          ;*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
6081          ;*      TRACK 2, SECTOR 25.  READ DATA BACK ONE SECTOR
6082          ;*      AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
6083          ;*      DID TAKE PLACE.
6084          ;*
6085          ;*****
6086 017402 000004          TST51: SCOPE

```



```

6143 017544 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6144
6145 017546 004437 042636  JSR    R4,GENCOM      ;COMPARE DATA
6146 017552 100000          100000
6147 017554 001000          1000          ;1000 WORDS
6148 017556 000413          BR     15          ;NO MISCOMPARES-SKIP
6149 017560 104015          ERROR 15          ;REPORT 1ST ERROR
6150
6151 017562 013700 001634  MOV    ERRMT,RO    ;GET ERROR LIMIT
6152 017566 005300 64$: DEC    RO          ;DECREMENT COUNT
6153 017570 001406          BEQ    65$         ;IF ZERO - EXIT
6154 017572 004437 042636  JSR    R4,GENCOM  ;CONTINUE DATA COMPARE
6155 017576 040000          40000
6156 017600 000402          BR     65$         ;NO MORE ERRORS - EXIT
6157 017602 104016          ERROR 16          ;REPORT NEXT ERROR
6158 017604 000770          BR     64$         ;LOOP
6159 017606
6160
6161 017606
6162
6163 1$: *****
6164 *TEST 52 TWO SECTOR READ DATA (PART 1)
6165 *
6166 * ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6167 * TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS
6168 * READ.
6169 *
6170 * NOTE: TWO SECTOR WRITE DATA (PART 1) MUST BE
6171 * EXECUTED BEFORE THIS TEST.
6172 * *****
6173 ST52: SCOPE
6174 MOV    #10.,$TIMES ;DO 10. ITERATIONS
6175 TSSINIT ;CLEAR SUBSYSTEM
6176 ERROR 3 ;BAD INIT ERROR
6177
6178 ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA (PART 1)
6179
6180 ;
6181 ; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA PART 1
6182 JSR    R4,GENCOM  ;GENERATE DATA
6183 15          ;PATTERN 15
6184 1000        ;1000 WORDS
6185
6186 JSR    R4,GENCOM  ;CLEAR Ibuff TO ALL ONES
6187 2007
6188 1000
6189
6190 JSR    R4,LRLOAD  ;LOAD "L" REGS
6191 RDDATA      ;RDDATA
6192 -1000       ;-1000 WORDS
6193 Ibuff       ;IBUFF IS BUFF ADDRESS
6194 .BYTE 0     ;SECTOR 0
6195 .BYTE 0     ;TRACK 0
6196 312         ;CYLINDER 312
6197
6198 TLOADRK     ;LOAD RK REGS
TWAT96       ;WAIT FOR INTERRUPT

```

```

6199 017664 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6200
6201 017666 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6202 017670 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6203
6204 017672 004437 042636    JSR R4,GENCOM   ;COMPARE DATA
6205 017676 100000          100000
6206 017700 001000          1000           ;1000 WORDS
6207 017702 000413          BR 1$          ;NO MISCOMPARES-SKIP
6208 017704 104015          ERROR 15
6209
6210 017706 013700 001634    MOV ERRLMT,RO  ;GET ERROR LIMIT
6211 017712 005300          DEC RO         ;DECREMENT COUNT
6212 017714 001406          BEQ 65$       ;IF ZERO - EXIT
6213 017716 004437 042636    JSR R4,GENCOM ;CONTINUE DATA COMPARE
6214 017722 040000          40000
6215 017724 000402          BR 65$        ;NO MORE ERRORS - EXIT
6216 017726 104016          ERROR 16      ;REPORT NEXT ERROR
6217 017730 000770          BR 64$        ;LOOP
6218 017732
6219
6220 017732          1$:
6221          ;*****
6222          ;TEST 53 TWO SECTOR READ DATA (PART 2)
6223          ;
6224          ; ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6225          ; TRACK 0, SECTOR 23. VERIFY THAT CORRECT DATA IS
6226          ; READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.
6227          ;
6228          ; NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE
6229          ; EXECUTED BEFORE THIS TEST.
6230          ;
6231          ;*****
6232          ;ST53: SCOPE
6233          MOV #10.,$TIMES ;DO 10. ITERATIONS
6234          TSSINIT ;CLEAR SUBSYSTEM
6235          ERROR 3 ;BAD INIT ERROR
6236          ;
6237          ;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)
6238          ;
6239          JSR R4,GENCOM ;GENERATE DATA
6240          16 ;PATTERN 16
6241          1000 ;1000 WORDS
6242          ;
6243          JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6244          2007
6245          1000
6246          JSR R4,LRLOAD ;LOAD "L" REGS
6247          RDDATA ;RDDATA
6248          -1000 ;-1000 WORDS
6249          Ibuff ;IBUFF IS BUFF ADDRESS
6250          .BYTE 23 ;SECTOR 23
6251          .BYTE 0 ;TRACK 0
6252          312 ;CYLINDER 312
6253          ;
6254          TLOADRK ;LOAD RK REGS

```

```

6255 020006 104430          TWAT96          ;WAIT FOR INTERRUPT
6256 020010 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6257
6258 020012 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6259 020014 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6260
6261 020016 004437 042636    JSR R4,GENCOM   ;COMPARE DATA
6262 020022 100000          100000
6263 020024 001000          1000           ;1000 WORDS
6264 020026 000413          BR 1$          ;NO MISCOMPARES-SKIP
6265 020030 104015          ERROR 1$       ;REPORT 1ST ERROR
6266
6267 020032 013700 001634    MOV ERRLMT,R0  ;GET ERROR LIMIT
6268 020036 005300          DEC R0         ;DECREMENT COUNT
6269 020040 001406          BEQ 65$       ;IF ZERO - EXIT
6270 020042 004437 042636    JSR R4,GENCOM   ;CONTINUE DATA COMPARE
6271 020046 040000          40000
6272 020050 000402          BR 65$        ;NO MORE ERRORS - EXIT
6273 020052 104016          ERROR 1$       ;REPORT NEXT ERROR
6274 020054 000770          BR 64$        ;LOOP
6275 020056
6276
6277 020056
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289 020056 000004          ;*****
6290 020060 012737 000012 001262  †ST54: SCOPE      ;*****
6291 020066 104416          MOV #10.,$TIMES ;DO 10. ITERATIONS
6292 020070 104003          TSSINIT        ;CLEAR SUBSYSTEM
6293
6294
6295 ; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)
6296 020072 004437 042636    JSR R4,GENCOM   ;GENERATE DATA
6297 020076 000002          2              ;PATTERN 2
6298 020100 000401          401           ;401 WORDS
6299
6300 020102 004437 042636    JSR R4,GENCOM   ;CLEAR Ibuff TO ALL ONES
6301 020106 002007          2007
6302 020110 001000          1000
6303
6304 020112 004437 036252    JSR R4,LRLOAD   ;LOAD "L" REGS
6305 020116 000121          RDDATA        ;RDDATA
6306 020120 177377          -401         ;-401 WORDS
6307 020122 070414          Ibuff        ;IBUFF IS BUFF ADDRESS
6308 020124 010           .BYTE 10       ;SECTOR 10
6309 020125 000           .BYTE 0        ;TRACK 0
6310 020126 000312          312          ;CYLINDER 312

```

```

6311
6312 020130 104417 TLOADRK ;LOAD RK REGS
6313 020132 104430 TWAT96 ;WAIT FOR INTERRUPT
6314 020134 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6315
6316 020136 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6317 020140 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6318
6319 020142 004437 042636 JSR R4,GENCOM ;COMPARE DATA
6320 020146 100000 100000 ;401 WORDS
6321 020150 000401 401 ;NO MISCOMPARES-SKIP
6322 020152 000413 BR 1$ ;PRINT FIRST ERROR
6323 020154 104015 ERROR 15
6324
6325 020156 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
6326 020162 005300 64$: DEC RO ;DECREMENT COUNT
6327 020164 001406 BEQ 65$ ;IF ZERO - EXIT
6328 020166 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
6329 020172 040000 40000
6330 020174 000402 BR 65$ ;NO MORE ERRORS - EXIT
6331 020176 104016 ERROR 16 ;REPORT NEXT ERROR
6332 020200 000770 BR 64$ ;LOOP
6333
6334 020202 65$:
6335 1$:
6336 *****
6337 *TEST 55 MID-TRANSFER SEEK ON READ (PART 1)
6338 *
6339 * ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6340 * TRACK 0, SECTOR 25. VERIFY THAT CORRECT DATA IS
6341 * READ AND A MID-TRANSFER SEEK DOES OCCUR.
6342 *
6343 * NOTE: MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
6344 * EXECUTED BEFORE THIS TEST.
6345 *****
6346 020202 000004 ST55: SCOPE
6347 020204 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
6348 020212 104416 TSSINIT ;CLEAR SUBSYSTEM
6349 020214 104003 ERROR 3 ;BAD INIT ERROR
6350
6351 ; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
6352 020216 004437 042636 JSR R4,GENCOM ;GENERATE DATA
6353 020222 000003 3 ;PATTERN 3
6354 020224 001000 1000 ;1000 WORDS
6355
6356 020226 004437 042636 JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6357 020232 002007 2007
6358 020234 001000 1000
6359
6360 020236 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
6361 020242 000121 RDDATA ;RDDATA
6362 020244 177000 -1000 ;-1000 WORDS
6363 020246 070414 Ibuff ;IBUFF IS BUFF ADDRESS
6364 020250 025 .BYTE 25 ;SECTOR 25
6365 020251 000 .BYTE 0 ;TRACK 0
6366 020252 000312 312 ;CYLINDER 312

```



```

6367
6368 020254 104417 TLOADRK ;LOAD RK REGS
6369 020256 104430 TWAT96 ;WAIT FOR INTERRUPT
6370 020260 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6371
6372 020262 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6373 020264 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6374
6375 020266 004437 042636 JSR R4,GENCOM ;COMPARE DATA
6376 020272 100000 100000 ;1000 WORDS
6377 020274 001000 BR 15 ;NO MISCOMPARES-SKIP
6378 020276 000413 ERROR 15 ;PRINT FIRST ERROR
6379 020300 104015
6380
6381 020302 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
6382 020306 005300 DEC RO ;DECREMENT COUNT
6383 020310 001406 BEQ 65$ ;IF ZERO - EXIT
6384 020312 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
6385 020316 040000 40000
6386 020320 000402 BR 65$ ;NO MORE ERRORS - EXIT
6387 020322 104016 ERROR 16 ;REPORT NEXT ERROR
6388 020324 000770 BR 64$ ;LOOP
6389 020326
6390 020326
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402 020326 000004 000012 001262 ST56: SCOPE
6403 020330 012737 MOV #10.,$TIMES ;DO 10. ITERATIONS
6404 020336 104416 TSSINIT ;CLEAR SUBSYSTEM
6405 020340 104003 ERROR 3 ;BAD INIT ERROR
6406
6407 ; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
6408 020342 004437 042636 JSR R4,GENCOM ;GENERATE DATA
6409 020346 000004 4 ;PATTERN 4
6410 020350 001000 1000 ;1000 WORDS
6411
6412 020352 004437 042636 JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6413 020356 002007 2007
6414 020360 001000 1000
6415
6416 020362 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
6417 020366 000121 RDDATA ;RDDATA
6418 020370 177000 -1000 ;-1000 WORDS
6419 020372 070414 Ibuff ;IBUFF IS BUFF ADDRESS
6420 020374 025 .BYTE 25 ;SECTOR 25
6421 020375 002 .BYTE 2 ;TRACK 2
6422 020376 000312 312 ;CYLINDER 312

```

```

6423
6424 020400 104417 TLOADRK ;LOAD RK REGS
6425 020402 104430 TWAT96 ;WAIT FOR INTERRUPT
6426 020404 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6427
6428 020406 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6429 020410 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6430
6431 020412 004437 042636 JSR R4,GENCOM ;COMPARE DATA
6432 020416 100000 100000 ;1000 WORDS
6433 020420 001000 1000 ;NO MISCOMPARES-SKIP
6434 020422 000413 BR 1$ ;REPORT FIRST ERROR
6435 020424 104015 ERROR 15
6436
6437 020426 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
6438 020432 005300 64$: DEC RO ;DECREMENT COUNT
6439 020434 001406 BEQ 65$ ;IF ZERO - EXIT
6440 020436 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
6441 020442 040000 40000
6442 020444 000402 BR 65$ ;NO MORE ERRORS - EXIT
6443 020446 104016 ERROR 16 ;REPORT NEXT ERROR
6444 020450 000770 BR 64$ ;LOOP
6445
6446 020452 1$:
6447 *****
6448 *TEST 57 CYLINDER ADDRESS OVERFLOW (PART 1)
6449 *****
6450 *
6451 * ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632
6452 * TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS
6453 * OVERFLOW ERROR DOES NOT OCCUR.
6454 * CYLINDER 1456 IS USED FOR THE RK07.
6455 *****
6456 020452 000004 TST57: SCOPE
6457 020454 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
6458 020462 104416 TSSINIT ;CLEAR SUBSYSTEM
6459 020464 104003 ERROR 3 ;BAD INIT ERROR
6460
6461 020466 013737 036476 020510 MOV LSTCYL,1$
6462 020474 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
6463 020500 000121 RDATA ;400 WORDS
6464 020502 177400 -400 ;IBUFF IS BUFF ADDR
6465 020504 070414 Ibuff ;SECTOR 25
6466 020506 025 .BYTE 25 ;TRK 2
6467 020507 002 .BYTE 2 ;CYL 632/1456
6468 020510 000000 1$: 0
6469
6470 020512 104417 TLOADRK ;LOAD RK REGS
6471 020514 104434 TWAT159 ;WAIT FOR INTERRUPT
6472 020516 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6473
6474 020520 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6475 020522 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6476 *****
6477 *TEST 60 CYLINDER ADDRESS OVERFLOW (PART 2)
6478 *****

```

E10

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 121
T60 CYLINDER ADDRESS OVERFLOW (PART 2)

SEQ 0121

6479				
6480				
6481				
6482				
6483				
6484				
6485	020524	000004		
6486	020526	012737	000012	001262
6487	020534	104416		
6488	020536	104003		
6489	020540	013737	036476	020562
6490	020546	004437	036252	
6491	020552	000121		
6492	020554	177377		
6493	020556	070414		
6494	020560	025		
6495	020561	002		
6496	020562	000000		
6497				
6498	020564	104417		
6499	020566	104434		
6500	020570	104002		
6501				
6502	020572	104422		
6503	020574	001000		
6504	020576	000000		
6505	020600	000000		
6506	020602	104004		
6507				
6508				
6509				
6510				
6511				
6512				
6513				
6514				
6515				
6516				
6517				
6518				
6519				
6520	020604	000004		
6521	020606	012737	000012	001262
6522	020614	012737	000312	001676
6523	020622	012737	020632	001110
6524	020630	005001		
6525	020632			
6526	020632	104416		
6527	020634	104003		
6528	020636	012737	010127	001600
6529	020644	053737	001720	001600
6530	020652	013737	001626	001610
6531	020660	012737	000074	001602
6532	020666	110137	001607	
6533	020672	012737	072414	001604
6534	020700	012737	000312	001614

```

;*      ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,
;*      TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS
;*      OVERFLOW ERROR DOES OCCUR.
;*      CYLINDER 1456 IS USED FOR THE RK07.
*****
↑ST60:  SCOPE
        MOV      #10.,$TIMES      ;;DO 10. ITERATIONS
        TSSINIT      ;;CLEAR SUBSYSTEM
        ERROR      3              ;;BAD INIT ERROR
        MOV      LSTCYL,1$
        JSR      R4,LRL0AD      ;;LOAD "L" REGS
        R0DATA
        -401              ;;401 WORDS
        Ibuff      ;;IBUFF IS BUFF ADDR
        .BYTE      25          ;;SEC 25
        .BYTE      2          ;;TRK 2
1$:      0                  ;;CYL 632/1456

        TLOADRK      ;;LOAD RK REGS
        TWAT159      ;;WAIT FOR INTERRUPT
        ERROR      2          ;;TO SLOW/NOT COMPLETE ERROR

        TCHKWE      ;;CHECK OPERATION WITH EXPECTED ERROR
        COERR        ;;CYLINDER OVERFLOW
        0
        0
        ERROR      4; OR 5,6,7  ;;REPORT ANY DISCREPANCIES

```

.SBTTL **18 BIT DATA TRANSFER TESTS

```

*****
*TEST 61      FORMAT IN 24 SECTOR FORMAT
*****
        FORMAT CYLINDER 312, TRACK 0, AND TRACK 1 FOR 24 SECTOR
        FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT
        OCCUR WITH WRITE HEADER ON IN READING DATA BUFFER
        AFTER READ HEADER.
*****

```

```

*****
↑ST61:  SCOPE
        MOV      #10.,$TIMES      ;;DO 10. ITERATIONS
        MOV      #312,REFMT      ;;SET REFORMAT SWITCH
        MOV      #1$,$LPERR      ;;SET LOCAL LOOP ON ERROR
        CLR      R1              ;;CLEAR R1 FOR TRACK COUNTER
1$:      TSSINIT      ;;CLEAR SUBSYSTEM
        ERROR      3              ;;BAD INIT ERROR
        MOV      #WRHEAD!CFMT,L.CS1 ;;SET UP FOR WRITE HEADER 24(8) SECTOR MODE
        BIS      DTYPE,L.CS1      ;;SET DRV TYP
        MOV      DRVNUM,L.CS2      ;;SET DRIVE NUMBER
        MOV      #74,L.WC          ;;SET WORD COUNT
        MOV      R1,L.DT          ;;LOAD TRACK ADDRESS
        MOV      #0BUFF,L.BA      ;;SET BUS ADDRESS
        MOV      #312,L.DCYL      ;;CYLINDER ADDRESS

```


G10

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 123
T61 FORMAT IN 24 SECTOR FORMAT

SEQ 0123

```

6591 021070
6592
6593
6594
6595
6596
6597
6598
6599
6600 021070 000004
6601 021072 012737 000012 001262
6602 021100 012737 000312 001676
6603 021106 104416
6604 021110 104003
6605
6606 021112 004437 042636
6607 021116 000013
6608 021120 000400
6609
6610 021122 004437 042636
6611 021126 002007
6612 021130 000400
6613
6614 021132 004437 036252
6615 021136 010123
6616 021140 177400
6617 021142 072414
6618 021144 000
6619 021145 000
6620 021146 000312
6621
6622 021150 104417
6623 021152 104430
6624 021154 104002
6625
6626 021156 104421
6627 021160 104004
6628
6629 021162 004437 036252
6630 021166 010121
6631 021170 177400
6632 021172 070414
6633 021174 000
6634 021175 000
6635 021176 000312
6636
6637 021200 104417
6638 021202 104424
6639 021204 104002
6640
6641 021206 104421
6642 021210 104004
6643
6644 021212 004437 042636
6645 021216 100000
6646 021220 000400

```

```

SS:
*****
*TEST 62      24 SECTOR FORMAT DATA TRANSFER (PART 1)
*
*      ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
*      TO CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK
*      AND MAKE SURE IT IS CORRECT.
*****
ST62:  SCOPE
        MOV      #10.,$TIMES      ;:DO 10. ITERATIONS
        MOV      #312,REFMT      ;:SET REFORMAT SWITCH
        TSSINIT                      ;:CLEAR SUBSYSTEM
        ERROR    3                ;:BAD INIT ERROR
        JSR      R4,GENCOM        ;:GENERATE DATA
        13                          ;:PATTERN 13
        400                          ;:400 WORDS
        JSR      R4,GENCOM        ;:CLEAR Ibuff TO ALL ONES
        2007
        400
        JSR      R4,LRLOAD        ;:LOAD "L" REGS
        WRDATA!CFMT                ;:WRDATA!CFMT
        -400                        ;:-400 WORDS
        OBUFF                      ;:OBUFF IS BUFF ADDRESS
        .BYTE    0                  ;:SECTOR 0
        .BYTE    0                  ;:TRACK 0
        312                          ;:CYLINDER 312
        TLOADRK                      ;:LOAD RK REGS
        TWAT96                      ;:WAIT FOR INTERRUPT
        ERROR    2                  ;:TO SLOW/NOT COMPLETE ERROR
        TCHKOP                      ;:CHECK OPERATION FOR ANY ERRORS
        ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
        JSR      R4,LRLOAD        ;:LOAD "L" REGS
        RDATA!CFMT                ;:RDATA!CFMT
        -400                        ;:-400 WORDS
        Ibuff                      ;:IBUFF IS BUFF ADDRESS
        .BYTE    0                  ;:SECTOR 0
        .BYTE    0                  ;:TRACK 0
        312                          ;:CYLINDER 312
        TLOADRK                      ;:LOAD RK REGS
        TWAT32                      ;:WAIT FOR INTERRUPT
        ERROR    2                  ;:TO SLOW/NOT COMPLETE ERROR
        TCHKOP                      ;:CHECK OPERATION FOR ANY ERRORS
        ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
        JSR      R4,GENCOM        ;:COMPARE DATA
        100000
        400                          ;:400 WORDS

```

H10

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 124
T62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

SEQ 0124

```
6647 021222 000413 BR 1$ ;NO MISCOMPARES-SKIP
6648 021224 104015 ERROR 15 ;REPORT 1ST ERROR
6649
6650 021226 013700 001634 MOV ERRMT,RO ;GET ERROR LIMIT
6651 021232 005300 64$: DEC RO ;DECREMENT COUNT
6652 021234 001406 BEQ 65$ ;IF ZERO - EXIT
6653 021236 004437 042636 JSR R4,GENCOM ;CONTINUE DATA COMPARE
6654 021242 040000 40000
6655 021244 000402 BR 65$ ;NO MORE ERRORS - EXIT
6656 021246 104016 ERROR 16 ;REPORT NEXT ERROR
6657 021250 000770 BR 64$ ;LOOP
6658 021252
6659
6660 021252 1$:
6661 *****
6662 :TEST 63 24 SECTOR FORMAT DATA TRANSFER (PART 2)
6663 *
6664 * LOAD A LOCATION WITH BAD PARITY. ISSUE A WRITE DATA OF
6665 * 400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0,
6666 * SECTOR 0 WITH BUFFER BEGINNING 112 WORDS BEFORE WORD
6667 * WITH BAD PARITY. MAKE SURE UNIBUS PARITY ERROR DOES NOT SET.
6668 * READ SECTOR BACK AND MAKE SURE IT IS CORRECT.
6669 *
6670 * NOTE: THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
6671 * EXISTS FOR SPECIFIED LOCATION.
6672 *
6673 *****
6674 :ST63: SCOPE
6675 021252 000004 MOV #10,$TIMES ;DO 10. ITERATIONS
6676 021254 012737 000012 001262 MOV #312,REFMT ;SET REFORMAT SWITCH
6677 021262 012737 000312 001676 BIT #PARPRE,OPTFLG ;PARITY OPTION PRESENT?
6678 021270 032737 000100 001664 BEQ 2$ ;YES-SKIP
6679
6680 021300 004437 042636 1$: JSR R4,GENCOM ;GENERATE DATA
6681 021304 000007 7 ;PATTERN 7
6682 021306 000400 400 ;400 WORDS
6683
6684 021310 032737 002000 001664 BIT #CP1170,OPTFLG ;TEST IF 11/70
6685 021316 001023 BNE 11$ ;YES - SKIP
6686 021320 012746 000340 MOV #PR7,-(SP) ;SET PRIORITY TO 7
6687 021324 012746 021332 MOV #10$,-(SP) ;SET ADDRESS
6688 021330 000002 RTI
6689 021332 10$:
6690
6691 021332 012777 000004 160332 MOV #BIT2,@CSRPTR ;SET WRONG PARITY WRITE
6692 021340 012737 100200 072640 MOV #100200,@BUFF+224 ;WRITE WITH BAD PARITY
6693 021346 012777 000001 160316 MOV #BIT0,@CSRPTR ;CLEAR WRONG PARITY, SET IE
6694
6695 021354 013746 001622 6$: MOV RKPRI,-(SP) ;RESTORE PRIORITY
6696 021360 012746 021366 MOV #11$,-(SP)
6697 021364 000002 RTI
6698 021366 11$:
6699 021366 104416 TSSINIT ;CLEAR SUBSYSTEM
6700 021370 104003 ERROR 3 ;BAD INIT ERROR
6701 021372 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
6702 021376 010123 WRDATA:CFMT ;WRDATA:CFMT
```

```

6703 021400 177400 -400 ; -400 WORDS
6704 021402 072414 OBUFF ; OBUFF IS BUFF ADDRESS
6705 021404 000 .BYTE 0 ; SECTOR 0
6706 021405 000 .BYTE 0 ; TRACK 0
6707 021406 000312 312 ; CYLINDER 312
6708
6709 021410 013704 001672 MOV CSRFTR,R4 ; LOAD R4 WITH CSR POINTER
6710 021414 005000 CLR R0 ; CLEAR R0 FOR COUNTING
6711 021416 012701 001662 MOV #INTSET,R1 ; LOAD R1 WITH POINTER INTERRUPT FLAG
6712 021422 012777 070400 160170 MOV #SPCHLR,ARKVEC ; SET INT VECTOR FOR RK611
6713 021430 012777 070410 160260 MOV #SPCPAR,AMMVECA ; AND PARITY ERROR
6714 021436 012737 021634 001264 MOV #4$,SESCAPE ; SET ESCAPE FOR ERROR
6715 021444 104417 TLOADRK ; LOAD RK REGS
6716 021446 032737 002000 001664 BIT #CP1170,OPTFLG ; TEST IF 11/70
6717 021454 001433 BEQ 45$ ; NO - SKIP
6718 021456 012737 000016 177746 MOV #16,177746 ; SET TO DISABLE CACHE
6719 021464 012777 000000 160216 MOV #0,AKWLADD ; DISABLE CLOCK INTERRUPTS
6720 021472 012777 000000 157444 MOV #0,ASTKS ; TURN OFF KEYBOARD INTERRUPTS
6721 021500 012714 170000 MOV #170000,(R4) ; SET TO FORCE PARITY ERROR
6722
6723 ;*****
6724 021504 105711 40$: TSTB (R1) ; LOOP TO WAIT FOR INTERRUPT. THE
6725 021506 003005 BGT 43$ ; CODE BETWEEN THE STARS IS SETUP SO ALL
6726 021510 005300 DEC R0 ; BYTES HAVE PARITY OF 1, HENCE NO PARITY
6727 021512 100774 BMI 40$ ; TRAPS ON AN 11/70. IF THIS CODE IS CHA-
6728 021514 000240 NOP ; NGED, REMEMBER ALL BYTES MUST HAVE AN
6729 021516 003372 BGT 40$ ; EVEN NUMBER OF BITS.
6730 021520 000240 NOP
6731 021522 005014 43$: CLR (R4) ; CLEAR ERROR FORCE
6732 ;*****
6733
6734 021524 005037 177746 CLR 177746 ; ENABLE CACHE
6735 021530 012777 000100 160152 MOV #BIT6,AKWLADD ; ENABLE CLOCK INTERRUPTS
6736 021536 012777 000100 157400 MOV #100,ASTKS ; TURN ON KEYBOARD INTERRUPTS
6737 021544 104430 45$: TWAT96 ; WAIT FOR INTERRUPT
6738 021546 000414 BR 46$ ; TO SLOW/NOT COMPLETE ERROR
6739 021550 032737 002000 001664 BIT #CP1170,OPTFLG ; TEST IF 11/70
6740 021556 001024 BNE 48$ ; YES - SKIP
6741 021560 005077 160106 CLR ACSRPTR ; CLEAR PARITY DETECT
6742 021564 005037 072640 CLR OBUFF+224 ; CLEAR BAD PARITY WORD
6743 021570 012777 000001 160074 MOV #1,ACSRPTR ; ENABLE PARITY DETECT
6744 021576 000414 BR 48$ ; SKIP
6745
6746 021600 032737 002000 001664 46$: BIT #CP1170,OPTFLG ; TEST IF 11/70
6747 021606 001007 BNE 47$ ; YES - SKIP
6748 021610 005077 160056 CLR ACSRPTR ; CLEAR PARITY DETECT
6749 021614 005037 072640 CLR OBUFF+224 ; CLEAR BAD PARITY WORD
6750 021620 012777 000001 160044 MOV #1,ACSRPTR ; ENABLE PARITY DETECT
6751 021626 104002 47$: ERROR 2 ; REPORT TO SLOW ERROR
6752
6753 021630 48$:
6754 021630 104421 TCHKOP ; CHECK OPERATION FOR ANY ERRORS
6755 021632 104004 ERROR 4 ; OR 5, 6, 7, 10 ; REPORT ALL ERRORS
6756 ; A UNIBUS PARITY ERROR REPORTED AT THIS
6757 ; TIME INDICATES CONTROLLER IS NOT
6758 ; PROPERLY TRANSFERRING DATA IN 18 BIT MODE

```

```

6759
6760
6761 021634 005037 001264
6762 021640 012777 035040 157752
6763 021646 004737 035120
6764 021652 012777 000100 157264
6765
6766 021660
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778 021660 000004
6779 021662 012737 000012 001262
6780 021670 012737 000312 001676
6781 021676 004737 035120
6782 021702 104416
6783 021704 104003
6784
6785 021706 004437 042636
6786 021712 000015
6787 021714 001000
6788
6789 021716 004437 042636
6790 021722 002007
6791 021724 001000
6792
6793 021726 004437 036252
6794 021732 010123
6795 021734 177000
6796 021736 072414
6797 021740 023
6798 021741 000
6799 021742 000312
6800
6801 021744 104417
6802 021746 104430
6803 021750 104002
6804
6805 021752 104421
6806 021754 104004
6807
6808 021756 004437 036252
6809 021762 010121
6810 021764 177000
6811 021766 070414
6812 021770 023
6813 021771 000
6814 021772 000312

```

```

4$: CLR $ESCAPE
MOV #INTHLR,ARKVEC ;RESET INT VECTOR FOR RK611
JSR PC,OPTTST
MOV #100,ASTKS ;ENABLE KEYBOARD INTERRUPTS

2$:
*****
*TEST 64 24 SECTOR FORMAT DATA TRANSFER (PART 3)
*
* ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT
* TO CYLINDER 312, TRACK 0, SECTOR 23. READ SECTOR BACK
* AND MAKE SURE IT IS CORRECT. MAKE SURE THAT MID-TRANSFER
* SEEK HAS TAKEN PLACE.
*****
†ST64: SCOPE
MOV #10,STIMES ;DO 10. ITERATIONS
MOV #312,REFMT ;SET REFORMAT SWITCH
JSR PC,OPTTST ;SET UP OPTIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

JSR R4,GENCOM ;GENERATE DATA
15 ;PATTERN 15
1000 ;1000 WORDS

JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
2007
1000

JSR R4,LRLOAD ;LOAD "L" REGS
WRDATA!CFMT ;WRDATA!CFMT
-1000 ;-1000 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 23 ;SECTOR 23
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

TLOADRK ;LOAD RK REGS
TWAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

JSR R4,LRLOAD ;LOAD "L" REGS
RDATA!CFMT ;RDATA!CFMT
-1000 ;-1000 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 23 ;SECTOR 23
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

```


K10

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 127
T64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

SEQ 0127

6815								
6816	021774	104417			TLOADRK			;LOAD RK REGS
6817	021776	104426			TWAT64			;WAIT FOR INTERRUPT
6818	022000	104002			ERROR 2			;TO SLOW/NOT COMPLETE ERROR
6819								
6820	022002	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
6821	022004	104004			ERROR 4	;OR 5, 6, 7, 10		;REPORT ALL ERRORS
6822								
6823	022006	004437	042636		JSR R4,GENCOM			;COMPARE DATA
6824	022012	100000			100000			
6825	022014	001000			1000			;1000 WORDS
6826	022016	000413			BR 1\$;NO MISCOMPARES-SKIP
6827	022020	104015			ERROR 15			;REPORT FIRST ERROR
6828								
6829	022022	013700	001634		MOV ERR,LMT,RO			;GET ERROR LIMIT
6830	022026	005300		64\$:	DEC RO			;DECREMENT COUNT
6831	022030	001406			BEQ 65\$;IF ZERO - EXIT
6832	022032	004437	042636		JSR R4,GENCOM			;CONTINUE DATA COMPARE
6833	022036	040000			40000			
6834	022040	000402			BR 65\$;NO MORE ERRORS - EXIT
6835	022042	104016			ERROR 16			;REPORT NEXT ERROR
6836	022044	000770			BR 64\$;LOOP
6837	022046			65\$:				
6838								
6839	022046			1\$:				

```

6840
6841 .SBTTL **SPECIAL DATA TRANSFER TESTS
6842
6843 :*****
6844 :*TEST 65 MULTI-SECTOR DATA TRANSFER AND BSE
6845 :*
6846 :* FORMAT CYLINDER 312, TRACK 0 IN 26 SECTOR FORMAT WITH
6847 :* SECTOR 1 MARKED BAD. ISSUE A WRITE DATA OF 1000 WORDS
6848 :* TO CYLINDER 312, TRACK 0, SECTOR 0. MAKE SURE BAD SECTOR
6849 :* ERROR SETS AND RKDA IS CORRECT. READ SECTOR 0 AND
6850 :* MAKE SURE IT IS CORRECT.
6851 :*
6852 :* ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312, TRACK 0,
6853 :* SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS AND THE
6854 :* PREVIOUS SECTOR IS LOADED CORRECTLY INTO MEMORY.
6855 :*
6856 :*****
6857 :*ST65: SCOPE
6858 :* MOV #10,$TIMES ;:DO 10. ITERATIONS
6859 :* MOV #312,REFMT ;:SET REFORMAT SWITCH
6860 :* TSSINIT ;:CLEAR SUBSYSTEM
6861 :* ERROR 3 ;:BAD INIT ERROR
6862
6863 :* JSR R4,LRLOAD ;:LOAD "L" REGS
6864 :* WRHEAD ;:WRHEAD
6865 :* -102 ;:-102 WORDS
6866 :* OBUFF ;:OBUFF IS BUFF ADDRESS
6867 :* .BYTE 0 ;:SECTOR 0
6868 :* .BYTE 0 ;:TRACK 0
6869 :* 312 ;:CYLINDER 312
6870

```

```

6871 022106 004437 042636 JSR R4,GENCOM ;BUILD HEADERS
6872 022112 000600 600
6873
6874 022114 042737 040000 072424 BIC #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD
6875 022122 042737 040000 072426 BIC #BIT14,OBUFF+12 ;CORRECT HURC
6876
6877 022130 104417 TLOADRK ;LOAD RK REGS
6878 022132 104431 TWAT112 ;WAIT FOR INTERRUPT
6879 022134 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6880
6881 022136 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6882 022140 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6883
6884 022142 004437 042636 JSR R4,GENCOM ;GENERATE DATA
6885 022146 000016 16 ;PATTERN 16
6886 022150 001000 1000 ;1000 WORDS
6887
6888 022152 004437 042636 JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6889 022156 002007 2007
6890 022160 001000 1000
6891
6892 022162 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
6893 022166 000123 WRDATA ;WRDATA
6894 022170 177000 -1000 ;-1000 WORDS
6895 022172 072414 OBUFF ;OBUFF IS BUFF ADDRESS
6896 022174 000 .BYTE 0 ;SECTOR 0
6897 022175 000 .BYTE 0 ;TRACK 0
6898 022176 000312 312 ;CYLINDER 312
6899
6900 022200 104417 TLOADRK ;LOAD RK REGS
6901 022202 104424 TWAT32 ;WAIT FOR INTERRUPT
6902 022204 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6903
6904 022206 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERR
6905 022210 000000 0
6906 022212 000100 BSERR ;BAD SECTOR ERROR
6907 022214 000000 0
6908 022216 104004 ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
6909 022220 005037 050360 CLR GRP4ER ;CLEAR GROUP 4 ERRORS
6910 022224 004437 040220 JSR R4,CHKCTS ;CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
6911 022230 032737 000020 050360 BIT #TAKERR,GRP4ER ;TRK IN ERROR?
6912 022236 001416 BEQ 1$ ;NO-SKIP
6913 022240 012737 054622 001450 MOV #EM13,EM10N ;"TRACK ADDRESS INCORRECT"
6914 022246 013737 050334 001202 MOV EXPTRK,$REG10 ;EXPECTED VALUE
6915 022254 013737 050346 001204 MOV REALTRK,$REG11 ;REAL VALUE
6916 022262 012737 051045 061434 MOV #OPER37,DF010A ;"AFTER WRITE DATA TERMINATED WITH BSE"
6917 022270 104010 ERROR 10
6918 022272 000527 BR 5$ ;EXIT
6919
6920 022274 032737 000040 050360 1$: BIT #SECERR,GRP4ER ;SECTOR IN ERROR?
6921 022302 001422 BEQ 3$ ;NO-SKIP
6922 022304 012737 054652 001450 MOV #EM14,EM10N ;"SECTOR ADDRESS INCORRECT"
6923 022312 012737 051045 061434 MOV #OPER37,DF010A ;"AFTER WRITE DATA ABORTED WITH BSE"
6924 022320 013737 050332 001202 MOV EXPSEC,$REG10 ;EXPECTED VALUE
6925 022326 013737 050350 001204 MOV REALSEC,$REG11 ;REAL VALUE
6926 022334 104010 ERROR 10

```

M10

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35MACY11 30A(1052) 10-JAN-78 09:57 PAGE 129
T65 MULTI-SECTOR DATA TRANSFER AND BSE

SEQ 0129

6927	022336	000505			BR	5\$;EXIT
6928	022340	104415			SCOP1			;LOCAL LOOP TO BEGINNING OF TEST
6929	022342	012737	022350	001110	MOV	#3\$, \$LPERR		;SET LOCAL LOOP ON ERROR
6930	022350						3\$:	
6931	022350	104416			TSSINIT			;CLEAR SUBSYSTEM
6932	022352	104003			ERROR	3		;BAD INIT ERROR
6933	022354	004437	036252		JSR	R4, LRLOAD		;LOAD "L" REGS
6934	022360	000121			RDDATA			;RDDATA
6935	022362	177400			-400			; -400 WORDS
6936	022364	070414			IBUFF			;IBUFF IS BUFF ADDRESS
6937	022366	000			.BYTE	0		;SECTOR 0
6938	022367	000			.BYTE	0		;TRACK 0
6939	022370	000312			312			;CYLINDER 312
6940								
6941	022372	104417			TLOADRK			;LOAD RK REGS
6942	022374	104424			TWAT32			;WAIT FOR INTERRUPT
6943	022376	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
6944								
6945	022400	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
6946	022402	104004			ERROR	4 ;OR 5, 6, 7, 10		;REPORT ALL ERRORS
6947								
6948	022404	004437	042636		JSR	R4, GENCOM		;COMPARE DATA
6949	022410	100000			100000			
6950	022412	000400			400			;400 WORDS
6951	022414	000413			BR	4\$;NO MISCOMPARES-EXIT
6952	022416	104015			ERROR	15		;REPORT FIRST ERROR
6953								
6954	022420	013700	001634		MOV	ERRLMT, R0		;GET ERROR LIMIT
6955	022424	005300			DEC	R0		;DECREMENT COUNT
6956	022426	001406			BEQ	65\$;IF ZERO - EXIT
6957	022430	004437	042636		JSR	R4, GENCOM		;CONTINUE DATA COMPARE
6958	022434	040000			40000			
6959	022436	000402			BR	65\$;NO MORE ERRORS - EXIT
6960	022440	104016			ERROR	16		;REPORT NEXT ERROR
6961	022442	000770			BR	64\$;LOOP
6962	022444						65\$:	
6963								
6964	022444	004437	042636		JSR	R4, GENCOM		;CLEAR IBUFF
6965	022450	002007			2007			
6966	022452	001000			1000			
6967								
6968	022454	004437	036252		JSR	R4, LRLOAD		;LOAD "L" REGS
6969	022460	000121			RDDATA			;RDDATA
6970	022462	177000			-1000			; -1000 WORDS
6971	022464	070414			IBUFF			;IBUFF IS BUFF ADDRESS
6972	022466	000			.BYTE	0		;SECTOR 0
6973	022467	000			.BYTE	0		;TRACK 0
6974	022470	000312			312			;CYLINDER 312
6975								
6976	022472	104417			TLOADRK			;LOAD RK REGS
6977	022474	104424			TWAT32			;WAIT FOR INTERRUPT
6978	022476	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
6979								
6980	022500	104422			TCHKWE			;CHECK OPERATION WITH EXPECTED ERROR
6981	022502	000000			0			
6982	022504	000100			BSERR			;BAD SECTOR ERROR

N10

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 130
T65 MULTI-SECTOR DATA TRANSFER AND BSE

SEQ 0130

6983	022506	000000			D			
6984	022510	104004			ERROR	4; OR 5,6,7		;REPORT ALL DISCREPANCIES
6985								
6986	022512	004437	042636		JSR	R4,GENCOM		;COMPARE DATA AGAIN
6987	022516	100000			100000			
6988	022520	000400			400			;400 WORDS
6989	022522	000413			BR	5\$;NO MISCOMPARES
6990	022524	104015			ERROR	15		;REPORT FIRST ERROR
6991								
6992	022526	013700	001634		MOV	ERRLMT,RO		;GET ERROR LIMIT
6993	022532	005300		66\$:	DEC	RO		;DECREMENT COUNT
6994	022534	001406			BEQ	67\$;IF ZERO - EXIT
6995	022536	004437	042636		JSR	R4,GENCOM		;CONTINUE DATA COMPARE
6996	022542	040000			40000			
6997	022544	000402			BR	67\$;NO MORE ERRORS - EXIT
6998	022546	104016			ERROR	16		;REPORT NEXT ERROR
6999	022550	000770			BR	66\$;LOOP
7000	022552							
7001								
7002	022552							
7003					5\$:			
7004					*****			
7005					*TEST 66	FORMAT TEST		
7006					* *			
7007					* *	FORMAT CYLINDER 312, TRACKS 0 AND 1 IN 26 SECTOR FORMAT.		
7008					* *	MAKE SURE NO ERRORS SET. READ SECTORS 0-25 AND MAKE		
7009					* *	SURE DATA CHECK DOES NOT OCCUR.		
7010					*****			
7011	022552	000004			†ST66:	SCOPE		
7012	022554	012737	000001	001262	MOV	#1,\$TIMES		;DO 1 ITERATION
7013	022552	005000			RO			;CLEAR TRACK COUNTER
7014	022564	012737	022572	001110	MOV	#1\$, \$LPERR		;SET LOCAL LOOP
7015								
7016	022572				1\$:			
7017	022572	104416			TSSINIT			;CLEAR SUBSYSTEM
7018	022574	104003			ERROR	3		;BAD INIT ERROR
7019								
7020	022576	013737	001626	001610	MOV	DRVNUM,L.CS2		;LOAD DRIVE NUMBER
7021	022604	012737	000127	001600	MOV	#WRHEAD,L.CS1		;LOAD WRITE HEADER
7022	022612	053737	001720	001600	BIS	DTYPE,L.CS1		;SET DRV TYP
7023	022620	110037	001607		RO,L.DT			;LOAD DESIRED TRACK FROM TRACK COUNTER
7024	022624	012737	072414	001604	MOV	#0BUFF,L.BA		;LOAD BUS ADDRESS
7025	022632	012737	177676	001602	MOV	#-102,L.WC		;WORD COUNT
7026	022640	012737	000312	001614	MOV	#312,L.DCYL		;CYLINDER
7027								
7028	022646	004437	042636		JSR	R4,GENCOM		;BUILD HEADER
7029	022652	001200			1200			;WITH BSE FLAGGED
7030								
7031	022654	104417			TLOADRK			;LOAD RK REGS
7032	022656	104431			TWAT112			;WAIT FOR INTERRUPT
7033	022660	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7034								
7035	022662	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7036	022664	104004			ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7037								
7038	022666	104415			SCOPI			;LOCAL LOOP TO 1\$

```

7039
7040 022670 005700          TST      RO          ;RO AT ZERO?
7041 022672 001002          BNE     2$          ;NO-EXIT
7042 022674 005200          INC     RO          ;BUMP COUNTER
7043 022676 000735          BR      1$          ;LOOP
7044 022700 005000          CLR     RO          ;CLEAR SECTOR COUNTER
7045 022702 012737 022710 001110 2$:  MOV     #113$,$LPERR ;SET LOCAL LOOP ON ERROR
7046
7047 022710          113$:
7048 022710 104416          TSSINIT          ;CLEAR SUBSYSTEM
7049 022712 104003          ERROR 3          ;BAD INIT ERROR
7050 022714          3$:
7051 022714 004437 036252          JSR     R4,LRLOAD ;LOAD "L" REGS
7052 022720 000121          RDDATA          ;RDDATA
7053 022722 177400          -400          ;-400 WORDS
7054 022724 070414          Ibuff          ;IBUFF IS BUFF ADDRESS
7055 022726 000          .BYTE 0          ;SECTOR 0
7056 022727 000          .BYTE 0          ;TRACK 0
7057 022730 000312          312          ;CYLINDER 312
7058
7059 022732 110037 001606          MOVVB  RO,L.DS    ;LOAD SECTOR COUNTER INTO DESIRED SECTOR
7060
7061 022736 104417          TLOADRK          ;LOAD RK REGS
7062 022740 104424          TWAT32          ;WAIT FOR INTERRUPT
7063 022742 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
7064
7065 022744 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
7066 022746 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7067
7068 022750 104415          SCOPI          ;LOCAL LOOP TO 3$ ON ERROR
7069
7070 022752 022700 000024          CMP     #24,RO    ;LAST SECTOR READ?
7071 022756 001402          BEQ     4$          ;YES-EXIT
7072 022760 005200          INC     RO          ;BUMP SECTOR COUNTER
7073 022762 000754          BR      3$          ;LOOP
7074
7075 022764 005037 001676          4$:  CLR     REFORMAT ;CLEAR REFORMAT SWITCH
7076
7077          .SBTTL **WRITE CHECK TESTS
7078
7079          ;*****
7080          ;*TEST 67 WRITE-CHECK WITH NO ERROR
7081          ;*
7082          ;* WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
7083          ;* DO A WRITE-CHECK OF 400 WORDS. MAKE SURE NO
7084          ;* ERROR OCCURS.
7085          ;*
7086          ;*****
7087 022770 000004          †ST67: SCOPE
7088 022772 012737 000012 001262          MOV     #10.,$TIMES ;DO 10. ITERATIONS
7089 023000 104416          TSSINIT          ;CLEAR SUBSYSTEM
7090 023002 104003          ERROR 3          ;BAD INIT ERROR
7091
7092 023004 004437 036252          JSR     R4,LRLOAD ;LOAD "L" REGS
7093 023010 000123          WRDATA          ;WRDATA
7094 023012 177400          -400          ;-400 WORDS

```

```

7095 023014 072414          OBUFF          ; OBUFF IS BUFF ADDRESS
7096 023016          000          .BYTE 0      ; SECTOR 0
7097 023017          000          .BYTE 0      ; TRACK 0
7098 023020 000312          312          ; CYLINDER 312
7099
7100 023022 004437 042636   JSR R4,GENCOM ; GENERATE DATA
7101 023026 000002          2          ; PATTERN 2
7102 023030 000400          400         ; 400 WORDS
7103
7104 023032 104417          TLOADRK      ; LOAD RK REGS
7105 023034 104430          TWAT96      ; WAIT FOR INTERRUPT
7106 023036 104002          ERROR 2     ; TO SLOW/NOT COMPLETE ERROR
7107
7108 023040 104421          TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
7109 023042 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7110
7111 023044 004437 036252   JSR R4,LRLOAD ; LOAD "L" REGS
7112 023050 000131          WRTCHK      ; WRTCHK
7113 023052 177400          -400        ; -400 WORDS
7114 023054 072414          OBUFF          ; OBUFF IS BUFF ADDRESS
7115 023056          000          .BYTE 0      ; SECTOR 0
7116 023057          000          .BYTE 0      ; TRACK 0
7117 023060 000312          312          ; CYLINDER 312
7118
7119 023062 104417          TLOADRK      ; LOAD RK REGS
7120 023064 104424          TWAT32      ; WAIT FOR INTERRUPT
7121 023066 104002          ERROR 2     ; TO SLOW/NOT COMPLETE ERROR
7122
7123 023070 104421          TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
7124 023072 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7125 *****
7126 *TEST 70          WRITE CHECK ERROR (PART 1)
7127 *
7128 *          WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
7129 *          WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME
7130 *          DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
7131 *          CONFIGURATIONS:
7132 *
7133 *          000001 000020 000400 010000
7134 *          000002 000040 001000 020000
7135 *          000004 000100 002000 040000
7136 *          000010 000200 004000 100000
7137 *
7138 *          MAKE SURE WRITE CHECK ERROR SET FOR EACH
7139 *          OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
7140 *          AND WORD COUNT IS CORRECT.
7141 *
7142 * *****
7143 023074 000004          TST70: SCOPE
7144 023076 012737 000012 001262 MOV #10, $TIMES ; DO 10. ITERATIONS
7145 023104 012700 000001 MOV #BIT0,RO ; SET LO ORDER BIT IN RO FOR
7146 ; CAUSING WRITE CHECK ERROR
7147 023110 104416          TSSINIT      ; CLEAR SUBSYSTEM
7148 023112 104003          ERROR 3     ; BAD INIT ERROR
7149 023114 004437 042636   JSR R4,GENCOM ; GENERATE DATA, ALL 0'S
7150 023120 000001          1

```

7151	023122	000400			400	
7152						
7153	023124	004437	036252		JSR R4,LRLOAD	;LOAD "L" REGS
7154	023130	000123			WRDATA	;WRDATA
7155	023132	177400			-400	; -400 WORDS
7156	023134	072414			OBUFF	;OBUFF IS BUFF ADDRESS
7157	023136	000			.BYTE 0	;SECTOR 0
7158	023137	000			.BYTE 0	;TRACK 0
7159	023140	000312			312	;CYLINDER 312
7160						
7161	023142	104417			TLOADRK	;LOAD RK REGS
7162	023144	104430			TWAT96	;WAIT FOR INTERRUPT
7163	023146	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
7164						
7165	023150	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
7166	023152	104004			ERROR 4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
7167						
7168	023154	004437	036252		JSR R4,LRLOAD	;LOAD "L" REGS
7169	023160	000131			WRTCHK	;WRTCHK
7170	023162	177400			-400	; -400 WORDS
7171	023164	072414			OBUFF	;OBUFF IS BUFF ADDRESS
7172	023166	000			.BYTE 0	;SECTOR 0
7173	023167	000			.BYTE 0	;TRACK 0
7174	023170	000312			312	;CYLINDER 312
7175						
7176	023172	104417			TLOADRK	;LOAD RK REGS
7177	023174	104424			TWAT32	;WAIT FOR INTERRUPT
7178	023176	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
7179						
7180	023200	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
7181	023202	104004			ERROR 4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
7182						
7183	023204	104415			SCOP1	;LOCAL LOOP ON WRITE CHECK
7184	023206	012737	023214	001110	MOV #1\$, \$LPERR	;SET LOCAL LOOP
7185	023214	010037	072634	1\$:	MOV R0, OBUFF+220	;CAUSE ERROR BIT IN BUFFER
7186	023220	104416			TSSINIT	;CLEAR SUBSYSTEM
7187	023222	104003			ERROR 3	;BAD INIT ERROR
7188	023224	004437	036252		JSR R4,LRLOAD	;LOAD "L" REGS
7189	023230	000131			WRTCHK	;WRTCHK
7190	023232	177400			-400	; -400 WORDS
7191	023234	072414			OBUFF	;OBUFF IS BUFF ADDRESS
7192	023236	000			.BYTE 0	;SECTOR 0
7193	023237	000			.BYTE 0	;TRACK 0
7194	023240	000312			312	;CYLINDER 312
7195						
7196	023242	104417			TLOADRK	;LOAD RK REGS
7197	023244	104424			TWAT32	;WAIT FOR INTERRUPT
7198	023246	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
7199						
7200	023250	104422			TCHKWE	;CHECK OPERATION WITH EXPECTED ERROR
7201	023252	000000			0	
7202	023254	000004			WCKERR	;WRITE CHECK ERROR
7203	023256	000000			0	
7204	023260	104004			ERROR 4; OR 5,6,7	;REPORT ALL DISCREPANCIES
7205						
7206	023262	104415			SCOP1	;LOCAL LOOP ON ERROR TO 1\$

NOTE: THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF THREE VALUES AND BE CORRECT. THE DIFFERENCE IS CAUSED BY WHEN THE WCE OCCURRED. IF IT OCCURRED ON THE FIRST WORD OF A DOUBLE NPR CYCLE, WC AND BA WILL BE TWO PAST WHERE THE ERROR ACTUALLY OCCURRED. IF WCE OCCURRED ON A SINGLE NPR CYCLE OR THE LAST NPR CYCLE OF A DOUBLE CYCLE, WC AND BA CONTENTS ARE ONE PAST THE ACTUAL WORD WHERE THE ERROR WAS. IN SOME CASES, WC AND BA CAN BE THREE PAST WHERE THE ERROR WAS AND STILL BE ACCEPTABLE.

```

7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218 023264 023727 001544 072636      CMP      T.BA,#OBUFF+222 ;CHECK BA HALT AT PROPER PLACE
7219 023272 001416                      BEQ      2$           ;YES-SKIP
7220 023274 101040                      BHI      6$           ;IF TO HI - SKIP
7221 023276 012737 054465 001450      MOV      #EM11,EM10N ;"INCORRECT BA"
7222 023304 012737 072636 001202      MOV      #OBUFF+222,$REG10 ;GOOD VALUE
7223 023312 013737 001544 001204      MOV      T.BA,$REG11 ;BAD VALUE
7224 023320 012737 051101 061434      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7225 023326 104010                      ERROR    10
7226
7227 023330 023727 001542 177511 2$:      CMP      T.WC,#-267   ;CHECK WORD COUNT AT CORRECT VALUE
7228 023336 001461                      BEQ      3$           ;YES-SKIP
7229 023340 101037                      BHI      7$           ;IF HIGHER SKIP
7230 023342 012737 054440 001450      MOV      #EM10,EM10N ;"INCORRECT WC"
7231 023350 012737 051101 061434      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7232 023356 012737 177511 001202      MOV      #-267,$REG10 ;GOOD VALUE
7233 023364 013737 001542 001204      MOV      T.WC,$REG11 ;ERROR VALUE
7234 023372 104010                      ERROR    10
7235 023374 000442                      BR       3$           ;EXIT
7236
7237 023376 023727 001544 072642 6$:      CMP      T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7238 023404 101415                      BLOS    7$           ;YES - SKIP
7239 023406 012737 054465 001450      MOV      #EM11,EM10N ;SET MESSAGE
7240 023414 012737 072642 001202      MOV      #OBUFF+226,$REG10 ;GOOD VALUE
7241 023422 013737 001544 001204      MOV      T.BA,$REG11 ;ERROR VALUE
7242 023430 012737 051101 061434      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7243 023436 104010                      ERROR    10
7244
7245 023440 023727 001542 177513 7$:      CMP      T.WC,#-265   ;TEST IF WORD COUNT AT HI SIDE
7246 023446 101415                      BLOS    3$           ;YES - SKIP
7247 023450 012737 054440 001450      MOV      #EM10,EM10N ;SET MESSAGE
7248 023456 012737 051101 061434      MOV      #OPER41,DF010A ;"WC ABORTED WITH WCE"
7249 023464 012737 177513 001202      MOV      #-265,$REG10 ;GOOD VALUE
7250 023472 013737 001542 001204      MOV      T.WC,$REG11 ;ERROR VALUE
7251 023500 104010                      ERROR    10
7252
7253 023502 104415                      3$:      SCOP1           ;LOCAL LOOP ON ERROR TO 1$
7254
7255 023504 032700 100000                      BIT      #BIT15,RO   ;BIT 15 SET?
7256 023510 001002                      BNE     4$           ;YES-EXIT
7257 023512 006300                      ASL     RO           ;SHIFT ERROR BIT
7258 023514 000637                      BR      1$           ;LOOP
7259
7260 023516
7261
7262
4$:
*****
;*TEST 71 WRITE CHECK ERROR (PART 2)

```



```

7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279 023516 000004
7280 023520 012737 000012 001262
7281 023526 012700 177776
7282
7283 023532 104416
7284 023534 104003
7285 023536 004437 042636
7286 023542 000007
7287 023544 000400
7288
7289 023546 004437 036252
7290 023552 000123
7291 023554 177400
7292 023556 072414
7293 023560 000
7294 023561 000
7295 023562 000312
7296
7297 023564 104417
7298 023566 104430
7299 023570 104002
7300
7301 023572 104421
7302 023574 104004
7303
7304 023576 004437 036252
7305 023602 000131
7306 023604 177400
7307 023606 072414
7308 023610 000
7309 023611 000
7310 023612 000312
7311
7312 023614 104417
7313 023616 104424
7314 023620 104002
7315
7316 023622 104421
7317 023624 104004
7318

```

```

;*
;* WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
;* IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0,
;* SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
;* ONE OF THE FOLLOWING CONFIGURATIONS:
;*
;* 177776 177757 177377 167777
;* 177775 177737 176777 157777
;* 177773 177677 175777 137777
;* 177767 177577 173777 077777
;*
;* MAKE SURE WRITE CHECK ERROR SET FOR EACH
;* OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
;* AND WORD COUNT IS CORRECT.
;*
;*****
;ST71: SCOPE
;MOV #10, $TIMES ;DO 10. ITERATIONS
;MOV #177776, R0 ;LOAD R0 FOR CAUSING WRITE CHECK ERROR
;
;TSSINIT ;CLEAR SUBSYSTEM
;ERROR 3 ;BAD INIT ERROR
;JSR R4, GENCOM ;GENERATE DATA
;7 ;ALL 1'S
;400 ;400 WORDS
;
;JSR R4, LRLOAD ;LOAD "L" REGS
;WRDATA ;WRDATA
;-400 ;-400 WORDS
;OBUFF ;OBUFF IS BUFF ADDRESS
;.BYTE 0 ;SECTOR 0
;.BYTE 0 ;TRACK 0
;312 ;CYLINDER 312
;
;TLOADRK ;LOAD RK REGS
;TWAT96 ;WAIT FOR INTERRUPT
;ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
;
;TCHKOP ;CHECK OPERATION FOR ANY ERRORS
;ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
;
;JSR R4, LRLOAD ;LOAD "L" REGS
;WRTCHK ;WRTCHK
;-400 ;-400 WORDS
;OBUFF ;OBUFF IS BUFF ADDRESS
;.BYTE 0 ;SECTOR 0
;.BYTE 0 ;TRACK 0
;312 ;CYLINDER 312
;
;TLOADRK ;LOAD RK REGS
;TWAT32 ;WAIT FOR INTERRUPT
;ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
;
;TCHKOP ;CHECK OPERATION FOR ANY ERRORS
;ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

```

G11

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 136
T71 WRITE CHECK ERROR (PART 2)

SEQ 0136

```

7319 023626 104415          SCOPI          ;LOCAL LOOP TO START OF TEST
7320 023630 012737 023636 001110  MOV          #1$, $LPERR          ;SET LOCAL LOOP
7321
7322 023636 010037 072634          1$:  MOV          R0, OBUFF+220      ;PUT WORD IN OBUFF TO CAUSE WCE
7323 023642 104416          TSSINIT          ;CLEAR SUBSYSTEM
7324 023644 104003          ERROR          3          ;BAD INIT ERROR
7325
7326 023646 004437 036252          JSR           R4, LRLOAD          ;LOAD "L" REGS
7327 023652 000131          WRTCHK          ;WRTCHK
7328 023654 177400          -400           ;-400 WORDS
7329 023656 072414          OBUFF          ;OBUFF IS BUFF ADDRESS
7330 023660          000           ;SECTOR 0
7331 023661          000           ;TRACK 0
7332 023662 000312          312           ;CYLINDER 312
7333
7334 023664 104417          TLOADRK          ;LOAD RK REGS
7335 023666 104424          TWAT32          ;WAIT FOR INTERRUPT
7336 023670 104002          ERROR          2          ;TO SLOW/NOT COMPLETE ERROR
7337
7338 023672 104422          TCHKWE          ;CHECK OPERATION WITH EXPECTED ERROR
7339 023674 000000          0
7340 023676 000004          WCKERR          ;WRITE CHECK ERROR
7341 023700 000000          0
7342 023702 104004          ERROR          4; OR 5,6,7      ;REPORT ALL DISCREPANCIES
7343
7344 023704 104415          SCOPI          ;LOCAL LOOP TO 1$
7345
7346          :
7347          :
7348          :
7349          :
7350          :
7351          :
7352          :
7353          :
7354          :
7355          :
7356 023706 023727 001544 072636          CMP           T.BA, #OBUFF+222    ;CHECK BA HALT AT PROPER PLACE
7357 023714 001416          BEQ           2$                    ;YES-SKIP
7358 023716 101040          BHI           6$                    ;IF TO HI - SKIP
7359 023720 012737 054465 001450          MOV          #EM11, EM10N          ;"INCORRECT BA"
7360 023726 012737 072636 001202          MOV          #OBUFF+222, $REG10    ;GOOD VALUE
7361 023734 013737 001544 001204          MOV          T.BA, $REG11          ;BAD VALUE
7362 023742 012737 051101 061434          MOV          #OPER41, DFO10A      ;"WRITE CHECK ABORTED WITH WCE"
7363 023750 104010          ERROR          10
7364
7365 023752 023727 001542 177511 2$:  CMP           T.WC, #-267          ;CHECK WORD COUNT AT CORRECT VALUE
7366 023760 001461          BEQ           3$                    ;YES-SKIP
7367 023762 101037          BHI           7$                    ;IF HIGHER - SKIP
7368 023764 012737 054440 001450          MOV          #EM10, EM10N          ;"INCORRECT WC"
7369 023772 012737 051101 061434          MOV          #OPER41, DFO10A      ;"WRITE CHECK ABORTED WITH WCE"
7370 024000 012737 177511 001202          MOV          #-267, $REG10        ;GOOD VALUE
7371 024006 013737 001542 001204          MOV          T.WC, $REG11         ;ERROR VALUE
7372 024014 104010          ERROR          10
7373 024016 000442          BR           3$                    ;EXIT
7374

```

H11

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 137
T71 WRITE CHECK ERROR (PART 2)

SEQ 0137

```

7375 024020 023727 001544 072642 6$: CMP T.BA,#OBUFF+226 ;TEST IF BA AT HI SIDE
7376 024026 101415 BLOS 7$ ;YES - SKIP
7377 024030 012737 054465 001450 MOV #EM11,EM10N ;SET MESSAGE
7378 024036 012737 072642 001202 MOV #OBUFF+226,$REG10 ;GOOD VALUE
7379 024044 013737 001544 001204 MOV T.BA,$REG11 ;ERROR VALUE
7380 024052 012737 051101 061434 MOV #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7381 024060 104010 ERROR 10
7382
7383 024062 023727 001542 177513 7$: CMP T.WC,#-265 ;TEST IF WORD COUNT AT HI SIDE
7384 024070 101415 BLOS 3$ ;YES - SKIP
7385 024072 012737 054440 001450 MOV #EM10,EM10N ;SET MESSAGE
7386 024100 012737 051101 061434 MOV #OPER41,DF010A ;"WC ABORTED WITH WCE"
7387 024106 012737 177513 001202 MOV #-265,$REG10 ;GOOD VALUE
7388 024114 013737 001542 001204 MOV T.WC,$REG11 ;ERROR VALUE
7389 024122 104010 ERROR 10
7390
7391 024124 104415 3$: SCOP1
7392
7393 024126 032700 100000 BIT #BIT15,RO ;BIT 15 SET? (ALL PATTERNS TESTED)
7394 024132 001002 BNE 4$ ;YES-EXIT
7395 024134 006300 ASL RO ;SHIFT FOR NEXT TEST
7396 024136 000637 BR 1$ ;LOOP
7397
7398 024140 4$:
7399 *****
7400 *TEST 72 WRITE CHECK OF PARTIAL SECTOR
7401 *
7402 * WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN
7403 * CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF
7404 * 110 WORDS MAKING SURE THE 111TH WORD IS
7405 * DIFFERENT THAN DATA ON DISK. MAKE SURE
7406 * WRITE CHECK ERROR DOES NOT SET.
7407 *
7408 *****
7409 ST72: SCOPE
7410 024140 000004 MOV #10.,$TIMES ;DO 10. ITERATIONS
7411 024142 012737 000012 001262 TSSINIT ;CLEAR SUBSYSTEM
7412 024150 104416 ERROR 3 ;BAD INIT ERROR
7413 024152 104003
7414 024154 004437 042636 JSR R4,GENCOM ;GENERATE DATA
7415 024160 000007 7 ;ALL 1'S
7416 024162 000400 400 ;400 WORDS
7417
7418 024164 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
7419 024170 000123 WRDATA ;WRDATA
7420 024172 177400 -400 ;-400 WORDS
7421 024174 072414 OBUFF ;OBUFF IS BUFF ADDRESS
7422 024176 000 .BYTE 0 ;SECTOR 0
7423 024177 000 .BYTE 0 ;TRACK 0
7424 024200 000312 312 ;CYLINDER 312
7425
7426 024202 104417 TLOADRK ;LOAD RK REGS
7427 024204 104430 TWAT96 ;WAIT FOR INTERRUPT
7428 024206 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7429
7430 024210 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS

```

```

7431 024212 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7432
7433 024214 005037 072636 CLR      OBUFF+222
7434
7435 024220 004437 036252 JSR      R4,LRLOAD      ;LOAD "L" REGS
7436 024224 000131      WRTCHK      ;WRTCHK
7437 024226 177670      -110        ;-110 WORDS
7438 024230 072414      OBUFF       ;OBUFF IS BUFF ADDRESS
7439 024232      000      .BYTE 0      ;SECTOR 0
7440 024233      000      .BYTE 0      ;TRACK 0
7441 024234 000312      312        ;CYLINDER 312
7442
7443 024236 104417      TLOADRK     ;LOAD RK REGS
7444 024240 104424      TWAT32     ;WAIT FOR INTERRUPT
7445 024242 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
7446
7447 024244 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
7448 024246 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7449

```

.SBTTL **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT

```

*****
*TEST 73      MAXIMUM DATA TRANSFER (PART 1)

```

```

*
* IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF
* THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO
* INSURE THE FORMAT IS CORRECT.
*
* ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH
* ONE SECTOR WRITES. ISSUE A SEEK TO CYLINDER 0, TRACK 0.
* ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000 WORDS
* TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE CONTROLLER
* TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS,
* DISK ADDRESS, BUS ADDRESS AND WORD COUNT. READ
* EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.
*
* NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
* IN THE FIRST 256 SECTORS ON THE PACK.

```

```

7471 024250 000004      TST73: SCOPE
7472 024252 012737 000002 001262 MOV      #2,$TIMES      ;:DO 2 ITERATIONS
7473 024260 032737 000400 001664 BIT      #FPFMT,OPTFLG ;:TEST IF FIRST PASS SWITCH SET
7474 024266 001043      BNE      24$          ;:YES - SKIP FORMAT
7475
7476 024270 105037 024313 CLR      21$          ;:CLEAR ADDRESS POINTERS
7477 024274 005037 024314 CLR      22$
7478
7479 024300 004437 036252 20$: JSR      R4,LRLOAD     ;:LOAD "L" REGISTERS
7480 024304 000127      WRHEAD     ;:WRITE HEADER
7481 024306 177676      -102      ;:102 WORDS
7482 024310 072414      OBUFF     ;:OBUFF IS BUFF ADDRESS
7483 024312      000      .BYTE 0      ;:SECTOR 0
7484 024313      000      .BYTE 0      ;:TRACK ADDRESS (VARIABLE)
7485 024314 000000      0        ;:CYLINDER 0 (VARIABLE)
7486

```

7487	024316	004437	042636			JSR	R4,GENCOM		;GO GENERATE HEADERS
7488	024322	001200				1200			;WITH BAD SECTOR ERRORS
7489									
7490	024324	104417				TLOADRK			;LOAD RK REGS
7491	024326	104431				TWAT112			;WAIT FOR INTERRUPT
7492	024330	104002				ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7493									
7494	024332	104421				TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7495	024334	104004				ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7496									
7497	024336	123727	024313	000002		CMPB	21\$,#2		;TEST IF LAST TRACK
7498	024344	001403				BEQ	23\$;YES - SKIP
7499	024346	105237	024313			INCB	21\$;ELSE BUMP TRACK
7500	024352	000752				BR	20\$;LOOP
7501									
7502	024354	105037	024313		23\$:	CLRB	21\$;CLEAR TRACK POINTER
7503	024360	023727	024314	000003		CMP	22\$,#3		;TEST IF LAST CYLINDER WRITTEN
7504	024366	001403				BEQ	24\$;YES - SKIP
7505	024370	005237	024314			INC	22\$;ELSE BUMP CYLINDER
7506	024374	000741				BR	20\$;LOOP
7507									
7508	024376	013737	024410	001110	24\$:	MOV	1\$, \$LPERR		;SET LOCAL LOOP ON ERROR
7509	024404	012703	000400			MOV	#400,R3		;SET COUNT FOR SECTOR CLEARING
7510	024410				1\$:				
7511	024410	104416				TSSINIT			;CLEAR SUBSYSTEM
7512	024412	104003				ERROR	3		;BAD INIT ERROR
7513	024414	004437	036252			JSR	R4,LRLOAD		;LOAD "L" REGS
7514	024420	000123				WRDATA			;WRDATA
7515	024422	177400				-400			; -400 WORDS
7516	024424	072414				OBUFF			;OBUFF IS BUFF ADDRESS
7517	024426	000				.BYTE	0		;SECTOR 0
7518	024427	000				.BYTE	0		;TRACK 0
7519	024430	000000				0			;CYLINDER 0
7520									
7521	024432	104417				TLOADRK			;LOAD RK REGS
7522	024434	104434				TWAT159			;WAIT FOR INTERRUPT
7523	024436	104002				ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7524									
7525	024440	104421				TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7526	024442	104004				ERROR	4 ;OR 5, 6, 7, 10		;REPORT ALL ERRORS
7527									
7528	024444	104415				SCOP1			;LOCAL LOOP ON ERROR TO 1\$
7529	024446	005303				DEC	R3		;DECREMENT COUNT
7530	024450	012762	072414	000004	2\$:	MOV	#OBUFF,RKBA(R2)		;SET BA
7531	024456	012762	177400	000002		MOV	#-400,RKWC(R2)		;AND WC AGAIN
7532	024464	005037	001662			CLR	INTSET		;CLEAR INTERRUPT FLAG
7533	024470	013762	001626	000010		MOV	DRVNUM,RKCS2(R2)		;SET DRIVE NUMBER
7534	024476	012737	000123	036502		MOV	#WRDATA,HOLD		
7535	024504	053737	001720	036502		BIS	DTYPE,HOLD		
7536	024512	013762	036502	000000		MOV	HOLD,RKCS1(R2)		;DO WRITE DATA
7537									
7538	024520	104425				TWAT48			;WAIT FOR INTERRUPT
7539	024522	104002				ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7540									
7541	024524	032762	000200	000014		BIT	#BSE,RKER(R2)		;BAD SECTOR ERROR?
7542	024532	001415				BEQ	3\$;NO-SKIP

K11

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 140
T73 MAXIMUM DATA TRANSFER (PART 1)

SEQ 0140

7543	024534	032737	000200	001664	BIT	#BSERPT,OPTFLG	;TEST IF BSE TO MANY HAS BEEN REPORTED
7544	024542	001007			BNE	5\$;YES - SKIP
7545	024544	052737	000200	001664	BIS	#BSERPT,OPTFLG	;SET FLAG
7546	024552	012737	053111	001360	MOV	#OPRO17,EMIN	;SET MESSAGE
7547	024560	104001			ERROR	1	; "FIRST 256 SECTOR NOT BSE FREE"
7548	024562	000137	025116		JMP	14\$;GO TO EXIT
7549							
7550	024566						
7551	024566	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7552	024570	104004			ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7553	024572	104415			SCOPI		;LOCAL LOOP TO IS (RESTART SECTOR CLEAR)
7554							
7555	024574	005303			DEC	R3	;DECREMENT COUNT
7556	024576	001324			BNE	2\$;LOOP IF NOT ZERO
7557							
7558	024600	004437	036252		JSR	R4,LRLOAD	;LOAD "L" REGS
7559	024604	000117			SEEK		;SEEK
7560	024606	000000			0		;0 WORDS
7561	024610	000000			0		;0 IS BUFF ADDRESS
7562	024612	000			.BYTE	0	;SECTOR 0
7563	024613	000			.BYTE	0	;TRACK 0
7564	024614	000000			0		;CYLINDER 0
7565							
7566	024616	104417			TLOADRK		;LOAD RK REGS
7567	024620	104423			TWAT16		;WAIT FOR INTERRUPT
7568	024622	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7569	024624	005037	001662		CLR	INTSET	;CLEAR FIRST INTERRUPT
7570	024630	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7571	024632	104004			ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
7572	024634	104427			TWAT80		;WAIT FOR SECOND INTERRUPT
7573	024636	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7574	024640	004437	042636		JSR	R4,GENCOM	;GENERATE DATA
7575	024644	004006			4006		;PATTERN 6, 1ST WORD REPEATED
7576	024646	000400			400		;400 WORDS
7577							
7578	024650						
7579	024650	104416			TSSINIT		;CLEAR SUBSYSTEM
7580	024652	104003			ERROR	3	;BAD INIT ERROR
7581	024654	004437	036252		JSR	R4,LRLOAD	;LOAD "L" REGS
7582	024660	000123			WRDATA		;WRDATA
7583	024662	000000			0		;0 WORDS
7584	024664	072414			OBUFF		;OBUFF IS BUFF ADDRESS
7585	024666	000			.BYTE	0	;SECTOR 0
7586	024667	000			.BYTE	0	;TRACK 0
7587	024670	000000			0		;CYLINDER 0
7588							
7589	024672	052737	000020	001610	BIS	#BAI,L.CS2	
7590							
7591	024700	104417			TLOADRK		;LOAD RK REGS
7592	024702	104437			TWAT8S		;WAIT FOR SECOND INTERRUPT
7593	024704	104002			ERROR	2	;ELSE REPORT TO SLOW/NOT COMPLETE ERROR
7594							
7595	024706						
7596	024706	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7597	024710	104004			ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7598							

7599	024712	104415			SCOP1		; INTERNAL LOOP ON ERROR TO 4\$
7600	024714	012703	000400		MOV	#400,R3	; SET COUNTER FOR READ-COMPARE LOOP
7601	024720	005037	024762		CLR	10\$; CLEAR SECTOR AND TRACK VALUES
7602	024724	005037	024764		CLR	12\$; CLEAR CYL VALUE
7603	024730	013737	024744	001110	MOV	8\$, \$LPERR	; SET LOCAL LOOP ON ERROR
7604	024736	042737	000020	001610	BIC	#BAI, L.CS2	; CLEAR BAI
7605							
7606	024744				8\$:		
7607	024744	104416			TSSINIT		; CLEAR SUBSYSTEM
7608	024746	104003			ERROR	3	; BAD INIT ERROR
7609	024750	004437	036252		9\$:	JSR R4, LRLOAD	; LOAD RK REGS
7610	024754	000121			RDDATA		; READ DATA
7611	024756	177400			-400		; 400 WORDS
7612	024760	070414			IBUFF		; INTO IBUFF
7613	024762	000			10\$:	.BYTE 0	; SECTOR (VARIABLE)
7614	024763	000			11\$:	.BYTE 0	; TRACK (VARIABLE)
7615	024764	000000			12\$:	0	; CYL (VARIABLE)
7616							
7617	024766	104417			TLOADRK		; LOAD RK REGS
7618	024770	104425			TWAT48		; WAIT FOR INTERRUPT
7619	024772	104002			ERROR	2	; TO SLOW/NOT COMPLETE ERROR
7620							
7621	024774	104421			TCHKOP		; CHECK OPERATION FOR ANY ERRORS
7622	024776	104004			ERROR	4 ;OR 5, 6, 7, 10	; REPORT ALL ERRORS
7623							
7624	025000	104415			SCOP1		; LOCAL LOOP ON ERROR TO 8\$
7625							
7626	025002	004437	042636		JSR	R4, GENCOM	; COMPARE DATA
7627	025006	100000			100000		
7628	025010	000400			400		; 400 WORDS
7629	025012	000413			BR	13\$; NO MISCOMPARE-EXIT LOOP
7630	025014	104015			ERROR	15	; REPORT FIRST ERROR
7631							
7632	025016	013700	001634		MOV	ERRLMT, R0	; GET ERROR LIMIT
7633	025022	005300			64\$:	DEC R0	; DECREMENT COUNT
7634	025024	001406			BEG	65\$; IF ZERO - EXIT
7635	025026	004437	042636		JSR	R4, GENCOM	; CONTINUE DATA COMPARE
7636	025032	040000			40000		
7637	025034	000402			BR	65\$; NO MORE ERRORS - EXIT
7638	025036	104016			ERROR	16	; REPORT NEXT ERROR
7639	025040	000770			BR	64\$; LOOP
7640	025042				65\$:		
7641							
7642	025042	104415			13\$:	SCOP1	; LOCAL LOOP TO 8\$
7643							
7644	025044	005303			DEC	R3	; DEC READ LOOP COUNT
7645	025046	001423			BEG	14\$; IF ZERO-EXIT
7646							
7647	025050	105237	024762		INCB	10\$; BUMP SECTOR
7648	025054	123727	024762	000026	CMPB	10\$, #26	; FINISHED WITH TRACK?
7649	025062	001332			BNE	9\$; NO-LOOP
7650	025064	105037	024762		CLRB	10\$; CLEAR SECTOR
7651	025070	105237	024763		INCB	11\$; BUMP TRACK
7652	025074	123727	024763	000003	CMPB	11\$, #3	; FINISHED WITH CYLINDER?
7653	025102	001322			BNE	9\$; NO-LOOP
7654	025104	105037	024763		CLRB	11\$; CLEAR TRACK

M11

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 142
T73 MAXIMUM DATA TRANSFER (PART 1)

SEG 0142

7655 025110 005237 024764
7656 025114 000715
7657
7658 025116

INC 12\$;BUMP CYL.
BR 9\$;LOOP

14\$:

*TEST 74 MAXIMUM DATA TRANSFER (PART 2)

7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675

ZERO OUT FIRST 256 SECTORS OF THE DISK WITH
200000 WORD WRITE. SEEK TO CYLINDER 632.
ISSUE A WRITE OF MAXIMUM DATA TRANSFER
200000 WORD WRITE. MAKE SURE CONTROLLER TIME
OUT IS NOT SET. CHECK CYLINDER ADDRESS
DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
SEEK TO CYLINDER 632. ISSUE A WRITE CHECK
OF 200000 WORDS. MAKE SURE NO ERROR SETS.

NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
IN THE FIRST 256 SECTORS ON THE PACK.
CYLINDER 1456 IS USED FOR RK07.

*ST74: SCOPE

7676 025116 000004
7677 025120 012737 000002 001262
7678 025126 104416
7679 025130 104003
7680 025132 012700 000620
7681 025136 005037 072414
7682
7683 025142 004437 036252
7684 025146 000123
7685 025150 000000
7686 025152 072414
7687 025154 000
7688 025155 000
7689 025156 000000
7690 025160 052737 000020 001610
7691 025166 104417
7692 025170 104434 1\$:
7693 025172 000401
7694 025174 000403
7695
7696 025176 005300 2\$:
7697 025200 001373
7698 025202 104002
7699
7700 025204 032762 000200 000014 3\$:
7701 025212 001415
7702
7703 025214 032737 000200 001664
7704 025222 001007
7705 025224 052737 000200 001664
7706 025232 012737 053111 001360
7707 025240 104001
7708 025242 000137 025610 12\$:
7709
7710 025246 4\$:

MOV #2,\$TIMES ;DO 2 ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
MOV #400.,RO ;SET COUNT FOR INTERRUPT WAIT
CLR OBUFF
JSR R4,LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
O ;O WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
O ;CYLINDER 0
BIS #BAI,L.CS2 ;SET BAI
TLOADRK ;LOAD RK REGS
TWAT159 ;WAIT FOR INTERRUPT
BR 2\$;NO INTERRUPT-SKIP
BR 3\$;INTERRUPT-SKIP
DEC RO ;DEC WAIT COUNTER
BNE 1\$;NO ZERO-LOOP
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
BIT #BSE,RKER(R2) ;DID BSE SET
BEQ 4\$;NO-SKIP
BIT #BSERPT,OPTFLG ;TEST IF TO MANY BAD SECTORS REPORTED
BNE 12\$;YES - SKIP
BIS #BSERPT,OPTFLG ;SET FLAG
MOV #OPRO17,EMIN ;SET MESSAGE
ERROR 1 ;"FIRST 256 SECTORS NOT BSE FREE"
JMP 11\$;EXIT


```

7711 025246 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7712 025250 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7713 025252 013737 036476 025302 MOV LSTCYL,13$
7714 025260 013737 036476 025456 MOV LSTCYL,14$
7715
7716 025266 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
7717 025272 000117 SEEK ;
7718 025274 000000 0 ;0 WORDS
7719 025276 000000 0 ;0 IS BUFF ADDR
7720 025300 000 .BYTE 0 ;SEC 0
7721 025301 000 .BYTE 0 ;TRK 0
7722 025302 000000 13$: 0 ;CYL 632/1456
7723
7724 025304 104417 TLOADRK ;LOAD RK REGS
7725 025306 104423 TWAT16 ;WAIT FOR INTERRUPT
7726 025310 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7727 025312 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
7728
7729 025316 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7730 025320 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7731
7732 025322 104427 TWAT80 ;WAIT FOR 2ND INTERRUPT
7733 025324 104002 ERROR 2
7734
7735 025326 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7736 025330 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7737
7738 025332 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
7739 025336 000105 CLEAR ;CLEAR
7740 025340 000000 0 ;0 WORDS
7741 025342 000000 0 ;0 IS BUFF ADDRESS
7742 025344 000 .BYTE 0 ;SECTOR 0
7743 025345 000 .BYTE 0 ;TRACK 0
7744 025346 000000 0 ;CYLINDER 0
7745
7746 025350 104417 TLOADRK ;LOAD RK REGS
7747 025352 104423 TWAT16 ;WAIT FOR INTERRUPT
7748 025354 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7749
7750 025356 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7751 025360 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7752
7753 025362 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
7754 025366 000123 WRDATA ;WRDATA
7755 025370 000000 0 ;0 WORDS
7756 025372 072414 OBUFF ;OBUFF IS BUFF ADDRESS
7757 025374 000 .BYTE 0 ;SECTOR 0
7758 025375 000 .BYTE 0 ;TRACK 0
7759 025376 000000 0 ;CYLINDER 0
7760
7761 025400 012737 135143 072414 MOV #135143,OBUFF ;SET WORD FOR OUTPUT
7762 025406 012700 000621 MOV #401.,R0 ;SET COUNT FOR INTERRUPT WAIT
7763 025412 052737 000020 001610 BIS #BAI,L.CS2 ;SET BUS ADDRESS INC INHIBIT
7764
7765 025420 104417 TLOADRK ;LOAD RK REGS
7766 025422 104434 5$: TWAT159 ;WAIT FOR INTERRUPT
    
```



```

7823 025566 104417          TLOADRK          ;LOAD RK REGS
7824 025570 104434          TWAT159         ;WAIT FOR INTERRUPT
7825 025572 000401          BR 9$          ;NO INTERRUPT-SKIP
7826 025574 000403          BR 10$         ;INTERRUPT-SKIP
7827
7828 025576 005300          9$: DEC R0      ;DEC WAIT COUNT
7829 025600 001373          BNE 8$         ;NOT ZERO-LOOP
7830 025602 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7831
7832 025604          10$: TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
7833 025604 104421          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7834 025606 104004
7835
7836 025610          11$: *****
7837          ;*****
7838          ;*TEST 75 CONTROLLER TIME OUT
7839          ;*
7840          ;*
7841          ;* SEEK TO CYLINDER 632. ISSUE A RECALIBRATE AND DO NOT
7842          ;* WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER
7843          ;* OF CYLINDER 0, TRACK 0. MAKE SURE CONTROLLER TIME
7844          ;* OUT SETS.
7845          ;* CYLINDER 1456 IS USED FOR THE RK07.
7846          ;*
7847          ;*****
7847 025610 000004          ST75: SCOPE
7848 025612 012737 000005 001262 MOV #5, $TIMES ;DO 5. ITERATIONS
7849 025620 104416          TSSINIT        ;CLEAR SUBSYSTEM
7850 025622 104003          ERROR 3        ;BAD INIT ERROR
7851 025624 013737 036476 025646 MOV LSTCYL, 1$
7852
7853 025632 004437 036252          JSR R4, LRLOAD ;LOAD "L" REGS
7854 025636 000117          SEEK
7855 025640 000000          0 ;0 WORDS
7856 025642 000000          0 ;0 IS BUFF ADDR
7857 025644 000          .BYTE 0 ;SEC 0
7858 025645 000          .BYTE 0 ;TRK 0
7859 025646 000000          1$: 0 ;CYL 632/1456
7860
7861 025650 104417          TLOADRK          ;LOAD RK REGS
7862 025652 104423          TWAT16         ;WAIT FOR INTERRUPT
7863 025654 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7864
7865 025656 104421          TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
7866 025660 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7867
7868 025662 005037 001662          CLR INTSET     ;CLEAR INTERRUPT FLAG
7869 025666 104427          TWAT80        ;WAIT FOR SECOND INTERRUPT
7870 025670 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7871 025672 104421          TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
7872 025674 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7873
7874 025676 004437 036252          JSR R4, LRLOAD ;LOAD "L" REGS
7875 025702 000105          CLEAR
7876 025704 000000          0 ;0 WORDS
7877 025706 000000          0 ;0 IS BUFF ADDRESS
7878 025710 000          .BYTE 0 ;SECTOR 0

```

```

7879 025711 000 .BYTE 0 ;TRACK 0
7880 025712 000000 0 ;CYLINDER 0
7881
7882 025714 104417 TLOADRK ;LOAD RK REGS
7883 025716 104423 TWAT16 ;WAIT FOR INTERRUPT
7884 025720 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7885
7886 025722 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7887 025724 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7888
7889 025726 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
7890 025732 000113 RECAL ;RECAL
7891 025734 000000 0 ;0 WORDS
7892 025736 000000 0 ;0 IS BUFF ADDRESS
7893 025740 000 .BYTE 0 ;SECTOR 0
7894 025741 000 .BYTE 0 ;TRACK 0
7895 025742 000000 0 ;CYLINDER 0
7896
7897 025744 104417 TLOADRK ;LOAD RK REGS
7898 025746 104423 TWAT16 ;WAIT FOR INTERRUPT
7899 025750 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7900
7901 025752 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
7902 025756 000125 RDHEAD ;RDHEAD
7903 025760 000000 0 ;0 WORDS
7904 025762 000000 0 ;0 IS BUFF ADDRESS
7905 025764 000 .BYTE 0 ;SECTOR 0
7906 025765 000 .BYTE 0 ;TRACK 0
7907 025766 000000 0 ;CYLINDER 0
7908
7909 025770 104417 TLOADRK ;LOAD RK REGS
7910 025772 104436 TWAT2S ;WAIT FOR INTERRUPT
7911 025774 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7912
7913 025776 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR
7914 026000 000000 0
7915 026002 000000 0
7916 026004 000002 CTOERR ;CONTROLLER TIME OUT
7917 026006 104004 ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
7918 026010 104416 TSSINIT ;CLEAR SUBSYSTEM
7919 026012 104003 ERROR 3 ;BAD INIT ERROR
7920 026014 005037 001662 CLR INTSET ;CLEAR INT FLAG
7921 026020 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INT ENABLE
7922 026026 104437 TWAT8S ;WAIT FOR SECOND INT
7923 026030 104002 ERROR 2
7924
7925
7926
7927
7928
7929
7930
7931
7932
7933
7934

```

.SBTTL **ERRORS DURING DATA TRANSFER

```

*****
*TEST 76 LIMIT DETECT ON DATA TRANSFER
*
* ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
* A SEEK TO CYLINDER 2 WITH BAD PARITY. ISSUE A DRIVE

```

```

7935      ;* CLEAR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
7936      ;* TRACK 0, HEAD 0. SEEK INCOMPLETE BECAUSE OF OUTER
7937      ;* LIMIT SHOULD BE THE ONLY ERROR SET.
7938      ;*****
7939 026032 000004      TST76: SCOPE
7940 026034 012737 000003 001262      MOV      #3.,$TIMES      ;;DO 3. ITERATIONS
7941
7942 026042 104416      TSSINIT      ;CLEAR SUBSYSTEM
7943 026044 104003      ERROR 3      ;BAD INIT ERROR
7944
7945 026046 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
7946 026052 000113      RECAL      ;RECAL
7947 026054 000000      0      ;0 WORDS
7948 026056 000000      0      ;0 IS BUFF ADDRESS
7949 026060 000      .BYTE 0      ;SECTOR 0
7950 026061 000      .BYTE 0      ;TRACK 0
7951 026062 000000      0      ;CYLINDER 0
7952
7953 026064 104417      TLOADRK      ;LOAD RK REGS
7954 026066 104423      TWAT16      ;WAIT FOR INTERRUPT
7955 026070 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
7956
7957 026072 005037 001662      CLR      INTSET      ;CLEAR INTERRUPT FLAG
7958 026076 104437      TWAT8S      ;WAIT FOR SECOND INTERRUPT
7959 026100 104002      ERROR 2
7960
7961 026102 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
7962 026104 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7963
7964 026106 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
7965 026112 000117      SEEK      ;SEEK
7966 026114 000000      0      ;0 WORDS
7967 026116 000000      0      ;0 IS BUFF ADDRESS
7968 026120 000      .BYTE 0      ;SECTOR 0
7969 026121 000      .BYTE 0      ;TRACK 0
7970 026122 000002      2      ;CYLINDER 2
7971 026124 012737 000020 001616      MOV      #PAT,L.MR1      ;SET EVEN PARITY BIT
7972 026132 104417      TLOADRK      ;LOAD RK REGS
7973 026134 104423      TWAT16      ;WAIT FOR INTERRUPT
7974 026136 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
7975
7976 026140 104416      TSSINIT      ;CLEAR SUBSYSTEM
7977 026142 104003      ERROR 3      ;BAD INIT ERROR
7978
7979 026144 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
7980 026150 000123      WRDATA      ;WRDATA
7981 026152 177400      -400      ;-400 WORDS
7982 026154 072414      OBUFF      ;OBUFF IS BUFF ADDRESS
7983 026156 000      .BYTE 0      ;SECTOR 0
7984 026157 000      .BYTE 0      ;TRACK 0
7985 026160 000001      1      ;CYLINDER 1
7986
7987 026162 104417      TLOADRK      ;LOAD RK REGS
7988 026164 104423      TWAT16      ;WAIT FOR INTERRUPT
7989 026166 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
7990

```

```

7991 026170 104422 TCHKWE ;CHECK OPERATION WITH ERROR
7992 026172 000002 SKIERR ;SEEK INCOMPLETE
7993 026174 000000 0
7994 026176 000000 0
7995 026200 104004 ERROR 4 ;OR 5,6,OR7 ;REPORT ALL DISCREPANCIES
7996
7997 026202 104416 TSSINIT ;CLEAR SUBSYSTEM
7998 026204 104003 ERROR 3 ;BAD INIT ERROR
7999
8000
8001 026206 004437 036252 3$: JSR R4,LRLOAD ;LOAD "L" REGS
8002 026206 000101 SELDRV ;SELDV
8003 026212 000000 0 ;0 WORDS
8004 026214 000000 0 ;0 IS BUFF ADDRESS
8005 026216 000000 .BYTE 0 ;SECTOR 0
8006 026220 000 .BYTE 0 ;TRACK 0
8007 026221 000 0 ;CYLINDER 0
8008 026222 000000
8009
8010 026224 012737 000001 001616 MOV #1,L.MR1 ;SET TO GET STATUS PAIR 1
8011 026232 104417 TLOADRK ;LOAD RK REGS
8012 026234 104423 TWAT16 ;WAIT FOR INTERRUPT
8013 026236 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8014
8015 026240 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8016 026242 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8017
8018 026244 032737 000040 001574 BIT #S.HDHM,T.MR2 ;TEST IF HEADS HOME
8019 026252 001755 BEQ 3$ ;NO - GET STATUS AGAIN
8020
8021 026254 104416 TSSINIT ;CLEAR SUBSYSTEM
8022 026256 104003 ERROR 3 ;BAD INIT ERROR
8023
8024 026260 012762 000100 000000 MOV #IE,RKCSI(R2) ;SET IE
8025
8026 026266 005037 001662 CLR INTSET ;CLEAR INT FLAG
8027 026272 104437 TWAT8S ;WAIT FOR SECOND INTERRUPT
8028 026274 000401 BR 1$
8029 026276 000404 BR 2$
8030
8031 026300 012737 060746 001372 1$: MOV #DH016,DH2N ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
8032 026306 104002 ERROR 2 ;ALLOWING HEADS TO RELOAD"
8033
8034
8035 026310 2$:
8036
8037
8038 ;*****
8039 ;*TEST 77 PROGRAMMING ERROR
8040 ;* ISSUE A SUBSYSTEM CLEAR. ISSUE
8041 ;* A READ DATA OF 400 WORDS ON CYLINDER 0,
8042 ;* TRACK 0, SECTOR 0. DURING READ ISSUE A
8043 ;* WRITE TO THE SPARE REGISTER. MAKE SURE
8044 ;* PROGRAMMING ERROR SETS.
8045 ;*****
8046 026310 000004 ST77: SCOPE
026312 012737 000012 001262 MOV #10.,$TIMES ;;DO 10. ITERATIONS

```

```

8047 026320 104416 TSSINIT ;CLEAR SUBSYSTEM
8048 026322 104003 ERROR 3 ;BAD INIT ERROR
8049
8050 026324 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8051 026330 000121 RDDATA ;RDDATA
8052 026332 177400 -400 ;-400 WORDS
8053 026334 070414 IBUFF ;IBUFF IS BUFF ADDRESS
8054 026336 000 .BYTE 0 ;SECTOR 0
8055 026337 000 .BYTE 0 ;TRACK 0
8056 026340 000000 0 ;CYLINDER 0
8057
8058 026342 104417 TLOADRK ;LOAD RK REGS
8059
8060 026344 012762 000001 000022 MOV #1,RKSPAR(R2) ;WRITE SPARE REGISTER
8061
8062 026352 104423 TWAT16 ;WAIT FOR INTERRUPT
8063 026354 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8064
8065 026356 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR
8066 026360 000000 0
8067 026362 000000 0
8068 026364 000020 PGERR ;PROG ERROR
8069 026366 104004 ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
8070
8071
8072
8073
8074
8075
8076
8077
8078
8079
8080
8081
8082
8083 026370 000004 TST100: SCOPE
8084 026372 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
8085 026400 104416 TSSINIT ;CLEAR SUBSYSTEM
8086 026402 104003 ERROR 3 ;BAD INIT ERROR
8087
8088 026404 004437 042636 JSR R4,GENCOM ;GENERATE DATA OF ALL ONES
8089 026410 000001 1
8090 026412 000400 400
8091
8092 026414 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8093 026420 000123 WRDATA ;WRDATA
8094 026422 177400 -400 ;-400 WORDS
8095 026424 072414 OBUFF ;OBUFF IS BUFF ADDRESS
8096 026426 000 .BYTE 0 ;SECTOR 0
8097 026427 000 .BYTE 0 ;TRACK 0
8098 026430 000000 0 ;CYLINDER 0
8099
8100 026432 104417 TLOADRK ;LOAD RK REGS
8101 026434 104430 TWAT96 ;WAIT FOR INTERRUPT
8102 026436 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

```

```

*****
*TEST 100 ECC HARD
* ISSUE A SUBSYSTEM CLEAR. ISSUE
* A WRITE DATA WORDS CONSISTING OF 177777 TO
* CYLINDER 0, TRACK 0, SECTOR 0. NOW WRITE
* ALL ZEROS TO CYLINDER 0, TRACK 0, SECTOR 0.
* DURING WRITE ISSUE CONTROLLER CLEAR. MAKE
* SURE PROGRAMMING ERROR IS RESET. NOW
* ISSUE A READ DATA TO CYLINDER 0, TRACK 0,
* HEAD 0 AND AN ECC HARD ERROR SHOULD SET.
*****

```

```

0103
0104 026440 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
0105 026442 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
0106
0107 026444 004437 042636 JSR R4,GENCOM ;GENERATE DATA OF ZEROS
0108 026450 000002 2
0109 026452 000400 400
0110
0111 026454 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
0112 026460 000123 WRDATA ;WRDATA
0113 026462 177630 -150 ;-150 WORDS
0114 026464 072414 OBUFF ;OBUFF IS BUFF ADDRESS
0115 026466 000 .BYTE 0 ;SECTOR 0
0116 026467 000 .BYTE 0 ;TRACK 0
0117 026470 000000 0 ;CYLINDER 0
0118
0119 026472 104417 TLOADRK ;START OPERATION
0120
0121 026474 005737 001662 1$: TST INTSET ;CHECK IF INTERRUPT HAS OCCURRED
0122 026500 001026 BNE 2$ ;YES - MUCH TO SOON. REPORT ERROR
0123 026502 005762 000002 TST RKWC(R2) ;TEST IF NPR'S DONE
0124 026506 001372 BNE 1$ ;NO - LOOP
0125
0126 026510 052762 100000 000000 BIS #CCLR,RKCS1(R2) ;CLEAR CONTROLLER (CROWBAR WRITE)
0127
0128 026516 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
0129 026522 000121 RDATA ;RDATA
0130 026524 177400 -400 ;-400 WORDS
0131 026526 070414 IBUFF ;IBUFF IS BUFF ADDRESS
0132 026530 000 .BYTE 0 ;SECTOR 0
0133 026531 000 .BYTE 0 ;TRACK 0
0134 026532 000000 0 ;CYLINDER 0
0135
0136 026534 104417 TLOADRK ;LOAD RK REGS
0137 026536 104425 TWAT48 ;WAIT FOR INTERRUPT
0138 026540 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
0139
0140 026542 104422 TCHKWE ;CHECK OPERATION WITH ERROR
0141 026544 000000 0
0142 026546 000003 DCKERR!ECHERR ;DATA CHECK AND ECC HARD
0143 026550 000000 0
0144 026552 104004 ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
0145
0146 026554 000402 BR 3$ ;SKIP TO EXIT
0147 026556 2$:
0148 026556 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
0149 026560 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
0150 026562 3$:

```

```

*****
*TEST 101 DRIVE TIMING ERROR
* ISSUE A SUBSYSTEM CLEAR. SEEK TO CYLINDER 632.
* ISSUE A RECALIBRATE BUT DO NOT WAIT FOR SECOND INTERRUPT.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A READ HEADER
* OF CYLINDER 0, TRACK 0. CLOCK THROUGH SEEK
*
```

```

0151
0152
0153
0154
0155
0156
0157
0158

```



```

8159      ;*      AND DRIVE CLEAR MESSAGES.  TURN OFF DIAGNOSTIC MODE.
8160      ;*      DRIVE TIMING ERROR SHOULD SET BECAUSE OF NO DATA
8161      ;*      TRANSISTIONS ON DATA LINE.
8162      ;*****
8163      026562 000004      †ST101: SCOPE
8164      026564 012737 000005 001262      MOV      #5.,$TIMES      ;;DO 5. ITERATIONS
8165
8166      026572 104416      TSSINIT      ;CLEAR SUBSYSTEM
8167      026574 104003      ERROR 3      ;BAD INIT ERROR
8168
8169      026576 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
8170      026602 000117      SEEK      ;SEEK
8171      026604 000000      0      ;0 WORDS
8172      026606 000000      0      ;0 IS BUFF ADDRESS
8173      026610      000      .BYTE 0      ;SECTOR 0
8174      026611      000      .BYTE 0      ;TRACK 0
8175      026612 000632      632      ;CYLINDER 632
8176
8177      026614 104417      TLOADRK      ;LOAD RK REGS
8178      026616 104423      TWAT16      ;WAIT FOR INTERRUPT
8179      026620 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8180      026622 005037 001662      CLR      INTSET      ;CLEAR INT FLAG
8181      026626 104430      TWAT96      ;WAIT FOR SECOND INTERRUPT
8182      026630 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8183
8184      026632 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
8185      026636 000113      RECAL      ;RECAL
8186      026640 000000      0      ;0 WORDS
8187      026642 000000      0      ;0 IS BUFF ADDRESS
8188      026644      000      .BYTE 0      ;SECTOR 0
8189      026645      000      .BYTE 0      ;TRACK 0
8190      026646 000000      0      ;CYLINDER 0
8191
8192      026650 104417      TLOADRK      ;LOAD RK REGS
8193      026652 104423      TWAT16      ;WAIT FOR INTERRUPT
8194      026654 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8195
8196      026656 004437 036252      JSR      R4,LRLOAD      ;LOAD "L" REGS
8197      026662 000125      RDHEAD      ;RDHEAD
8198      026664 000000      0      ;0 WORDS
8199      026666 000000      0      ;0 IS BUFF ADDRESS
8200      026670      000      .BYTE 0      ;SECTOR 0
8201      026671      000      .BYTE 0      ;TRACK 0
8202      026672 000000      0      ;CYLINDER 0
8203      026674 012737 000040 001616      MOV      #DMD,L.MR1      ;SET DIAG MODE
8204      026702 104417      TLOADRK      ;LOAD RK REGS
8205
8206      026704 004437 036674      JSR      R4,MCLOCK      ;CLOCK CONTROLLER THROUGH SEEK
8207      026710 001062      1062      ;AND CLEAR TO READ
8208
8209      026712 005062 000026      CLR      RKMR1(R2)      ;RESET DIAG MODE, LET RD HDRS COMPLETE
8210
8211      026716 104424      TWAT32      ;WAIT FOR INTERRUPT
8212      026720 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8213      026722 104422      TCHKWE      ;CHECK OPERATION WITH EXP ERROR
8214      026724 010000      DTERR      ;DRIVE TIMING ERROR

```

```

8215 026726 000000 0
8216 026730 000000 0
8217 026732 104004 ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
8218
8219 026734 104416 1$:
8220 026734 104003 TSSINIT ;CLEAR SUBSYSTEM
8221 026736 104003 ERROR 3 ;BAD INIT ERROR
8222 026740 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
8223 026746 005037 001662 CLR INTSET ;CLEAR INT FLAG
8224
8225 026752 104437 TWATBS ;WAIT FOR INTERRUPT FOR END OF RECAL
8226 026754 104002 ERROR 2
8227
8228 026756 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8229 026762 000105 CLEAR ;CLEAR
8230 026764 000000 0 ;0 WORDS
8231 026766 000000 0 ;0 IS BUFF ADDRESS
8232 026770 000 .BYTE 0 ;SECTOR 0
8233 026771 000 .BYTE 0 ;TRACK 0
8234 026772 000000 0 ;CYLINDER 0
8235
8236 026774 104417 TLOADRK ;LOAD RK REGS
8237 026776 104423 TWAT16 ;WAIT FOR INTERRUPT
8238 027000 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8239 027002 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8240 027004 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8241
8242 .SBTTL **ERROR FORCING IN DRIVE
8243
8244 *****
8245 *TEST 102 INITIALIZE CLEARING SACK
8246 *
8247 * ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
8248 * DRIVE. ISSUE A SUBSYSTEM CLEAR. PUT CONTROLLER IN
8249 * DIAGNOSTIC MODE. ISSUE A SELECT COMMAND WITH
8250 * MESSAGE ID = 3 AND DRIVE SELECTED = 0. CLOCK THROUGH
8251 * PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE. MAKE
8252 * SURE UNIT FIELD ERROR DOES NOT SET.
8253 *
8254 *****
8255 †ST102: SCOPE
8256 027006 000004 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
8257 027010 012737 TSSINIT ;CLEAR SUBSYSTEM
8258 027016 104416 000003 ERROR 3 ;BAD INIT ERROR
8259
8260 027022 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8261 027026 000101 SELDRV ;SELDV
8262 027030 000000 0 ;0 WORDS
8263 027032 000000 0 ;0 IS BUFF ADDRESS
8264 027034 000 .BYTE 0 ;SECTOR 0
8265 027035 000 .BYTE 0 ;TRACK 0
8266 027036 000000 0 ;CYLINDER 0
8267
8268 027040 104417 TLOADRK ;LOAD RK REGS
8269 027042 104423 TWAT16 ;WAIT FOR INTERRUPT
8270 027044 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

```

```

8271
8272 027046 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8273 027050 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8274
8275 027052 104416 TSSINIT ;CLEAR SUBSYSTEM
8276 027054 104003 ERROR 3 ;BAD INIT ERROR
8277
8278 027056 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8279 027062 000101 SELDRV ;SELDV
8280 027064 000000 0 ;0 WORDS
8281 027066 000000 0 ;0 IS BUFF ADDRESS
8282 027070 000 ;SECTOR 0
8283 027071 000 ;TRACK 0
8284 027072 000000 0 ;CYLINDER 0
8285 027074 012737 000043 001616 MOV #3!DMD,L.MR1 ;SET DIAG MODE AND MESSAGE PAIR 3
8286 027102 005037 001610 CLR L.CS2 ;SELECT DRIVE 0
8287
8288 027106 104417 TLOADRK ;LOAD RK REGS
8289
8290 027110 004437 036674 JSR R4,MCLOCK ;CLOCK THROUGH PHASE ADDRESS 6
8291 027114 001027 1027
8292
8293 027116 042762 000040 000026 BIC #DMD,RKMR1(R2) ;CLEAR MAINTENANCE MODE
8294
8295 027124 104424 TWAT32 ;WAIT FOR INTERRUPT
8296 027126 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
8297
8298 027130 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8299 027132 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8300 *****
8301 *TEST 103 DRIVE OFF TRACK *****
8302 *
8303 * ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
8304 * OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC
8305 * MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8306 * TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR
8307 * MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK
8308 * SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILIABLE DRIVES.
8309 *
8310 *****
8311 TST103: SCOPE
8312 027134 000004 MOV #5.,$TIMES ;DO 5. ITERATIONS
8313 027136 012737 000005 001262 TSSINIT ;CLEAR SUBSYSTEM
8314 027144 104416 ERROR 3 ;BAD INIT ERROR
8315 027146 104003
8316 027150 004437 036252 JSR R4,LRLOAD ;LOAD "L" REGS
8317 027154 000113 RECAL ;RECAL
8318 027156 000000 0 ;0 WORDS
8319 027160 000000 0 ;0 IS BUFF ADDRESS
8320 027162 000 ;SECTOR 0
8321 027163 000 ;TRACK 0
8322 027164 000000 0 ;CYLINDER 0
8323
8324 027166 104417 TLOADRK ;LOAD RK REGS
8325 027170 104423 TWAT16 ;WAIT FOR INTERRUPT
8326 027172 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

```

8327	027174	005037	001662	CLR	INTSET		;CLEAR INTERRUPT FLAG
8328							
8329	027200	104437		TWAT8S			;WAIT FOR INTERRUPT #2
8330	027202	104002		ERROR	2		
8331							
8332	027204	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
8333	027210	000105		CLEAR			;CLEAR
8334	027212	000000		0			;0 WORDS
8335	027214	000000		0			;0 IS BUFF ADDRESS
8336	027216	000		.BYTE	0		;SECTOR 0
8337	027217	000		.BYTE	0		;TRACK 0
8338	027220	000000		0			;CYLINDER 0
8339							
8340	027222	104417		TLOADRK			;LOAD RK REGS
8341	027224	104423		TWAT16			;WAIT FOR INTERRUPT
8342	027226	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8343							
8344	027230	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8345	027232	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8346							
8347	027234	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
8348	027240	000115		OFFSET			;OFFSET
8349	027242	000000		0			;0 WORDS
8350	027244	000000		0			;0 IS BUFF ADDRESS
8351	027246	000		.BYTE	0		;SECTOR 0
8352	027247	000		.BYTE	0		;TRACK 0
8353	027250	000000		0			;CYLINDER 0
8354	027252	112737	000060 001612	MOVB	#60,L.ASOF		;SET OFFSET AT +1200
8355							
8356	027260	104417		TLOADRK			;LOAD RK REGS
8357	027262	104423		TWAT16			;WAIT FOR INTERRUPT
8358	027264	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8359							
8360	027266	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8361	027270	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8362							
8363	027272	005037	001662	CLR	INTSET		;CLEAR INT FLAG
8364							
8365	027276	104424		TWAT32			;WAIT FOR INT #2
8366	027300	104002		ERROR	2		
8367							
8368	027302	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8369	027304	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8370							
8371	027306	004437	036252	JSR	R4,LRLOAD		;LOAD "L" REGS
8372	027312	000105		CLEAR			;CLEAR
8373	027314	000000		0			;0 WORDS
8374	027316	000000		0			;0 IS BUFF ADDRESS
8375	027320	000		.BYTE	0		;SECTOR 0
8376	027321	000		.BYTE	0		;TRACK 0
8377	027322	000000		0			;CYLINDER 0
8378							
8379	027324	104417		TLOADRK			;LOAD RK REGS
8380	027326	104423		TWAT16			;WAIT FOR INTERRUPT
8381	027330	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8382							

```

8383 027332 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8384 027334 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8385
8386 027336 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
8387 027342 000023 23 ;WRITE DATA W/O INTERRUPT ENABLE
8388 027344 177777 -1 ;-1 WORDS
8389 027346 072414 OBUFF ;OBUFF IS BUFF ADDRESS
8390 027350 000 .BYTE 0 ;SECTOR 0
8391 027351 000 .BYTE 0 ;TRACK 0
8392 027352 000000 0 ;CYLINDER 0
9393 027354 012737 000040 001616 MOV #DMD,L.MR1 ;SET DIAGNOSTIC MODE
8394
8395 027362 104417 TLOADRK
8396
8397 027364 004437 036674 JSR R4,MCLOCK ;CLOCK THROUGH SEEK & DRIVE CLEAR
8398 027370 001064 1064
8399
8400 027372 005062 000026 CLR RKMR1(R2) ;CLEAR DIAGNOSTIC MODE
8401 027376 104426 TWAT64 ;WAIT FOR INTERRUPT
8402 027400 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
8403 027406 104423 TWAT16 ;STALL FOR 16 MS
8404 027410 000240 NOP
8405 027412 000240 NOP
8406
8407 027414 004437 036252 JSR R4,LRLoad ;LOAD "L" REGS
8408 027420 000101 SELDRV ;SELDV
8409 027422 000000 0 ;0 WORDS
8410 027424 000000 0 ;0 IS BUFF ADDRESS
8411 027426 000 .BYTE 0 ;SECTOR 0
8412 027427 000 .BYTE 0 ;TRACK 0
8413 027430 000000 0 ;CYLINDER 0
8414 027432 005037 001616 CLR L.MR1 ;RESET DIAG MODE
8415
8416 027436 104417 TLOADRK ;LOAD RK REGS
8417 027440 104423 TWAT16 ;WAIT FOR INTERRUPT
8418 027442 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8419
8420 027444 104422 TCHKWE ;CHECK OPERATION WITH ERROR EXPECTED
8421 027446 000400 DROTERR ;DRIVE OFF TRACK
8422 027450 000000 0
8423 027452 000000 0
8424 027454 104004 ERROR 4 ;OR 5,6,7 ;REPORT ANY DISCREPANCIES

```

```

*****
*TEST 104 FILE UNSAFE
*
* ISSUE A SUBSYSTEM CLEAR. ISSUE A RECLAIBRATE. ISSUE
* A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
* FORMAT. DO A SELECT COMMAND IN 26 SECTOR FORMAT.
* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE
* HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
* FORMAT. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
* SIMULATE INDEX PULSE. TURN OFF DIAGNOSTIC MODE. FILE
* UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
* THROUGH SECTOR PULSE. REPEAT FOR ALL AVAILIABLE DRIVES.
*****

```

```

8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438

```

```

8439 027456 000004
8440 027460 012737 000005 001262 TST104: SCOPE
8441 027466 012737 177777 001676 MOV #5, $TIMES ;:DO 5. ITERATIONS
8442 027474 104416 TSSINIT ;:SET REFORMAT SWITCH
8443 027476 104003 ERROR 3 ;:CLEAR SUBSYSTEM
8444 ;:BAD INIT ERROR
8445 027500 004437 036252 JSR R4, LRLOAD ;:LOAD "L" REGS
8446 027504 000113 RECAL ;:RECAL
8447 027506 000000 0 ;:0 WORDS
8448 027510 000000 0 ;:0 IS BUFF ADDRESS
8449 027512 000 .BYTE 0 ;:SECTOR 0
8450 027514 000 .BYTE 0 ;:TRACK 0
8451 027516 000012 12 ;:CYLINDER 12
8452
8453 027516 104417 TLOADRK ;:LOAD RK REGS
8454 027520 104423 TWAT16 ;:WAIT FOR INTERRUPT
8455 027522 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
8456
8457 027524 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
8458 027526 104004 ERROR 4 ;OR 5, 6, 7 ;:REPORT ALL ERRORS
8459
8460 027530 005037 001662 CLR INTSET ;:CLEAR INT FLAG
8461 027534 104437 TWAT8S ;:WAIT FOR SECOND INT
8462 027536 104002 ERROR 2
8463
8464 027540 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
8465 027542 104004 ERROR 4 ;OR 5, 6, 7 ;:REPORT ALL ERRORS
8466
8467 027544 004437 036252 JSR R4, LRLOAD ;:LOAD "L" REGS
8468 027550 000105 CLEAR ;:CLEAR
8469 027552 000000 0 ;:0 WORDS
8470 027554 000000 0 ;:0 IS BUFF ADDRESS
8471 027556 000 .BYTE 0 ;:SECTOR 0
8472 027557 000 .BYTE 0 ;:TRACK 0
8473 027560 000012 12 ;:CYLINDER 12
8474
8475 027562 104417 TLOADRK ;:LOAD RK REGS
8476 027564 104423 TWAT16 ;:WAIT FOR INTERRUPT
8477 027566 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
8478
8479 027570 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
8480 027572 104004 ERROR 4 ;OR 5, 6, 7 ;:REPORT ALL ERRORS
8481
8482 027574 004437 036252 JSR R4, LRLOAD ;:LOAD "L" REGS
8483 027600 010125 RDHEAD:CFMT ;:RDHEAD:CFMT
8484 027602 000000 0 ;:0 WORDS
8485 027604 000000 0 ;:0 IS BUFF ADDRESS
8486 027606 000 .BYTE 0 ;:SECTOR 0
8487 027607 000 .BYTE 0 ;:TRACK 0
8488 027610 000012 12 ;:CYLINDER 12
8489
8490 027612 104417 TLOADRK ;:LOAD RK REGS
8491 027614 104424 TWAT32 ;:WAIT FOR INTERRUPT
8492 027616 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
8493
8494 027620 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS

```

8495	027622	104004			ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8496							
8497	027624	004437	036252		JSR	R4,LRLOAD	;LOAD "L" REGS
8498	027630	000101			SELDRV		;SELDRV
8499	027632	000000			0		;0 WORDS
8500	027634	000000			0		;0 IS BUFF ADDRESS
8501	027636	000			.BYTE	0	;SECTOR 0
8502	027637	000			.BYTE	0	;TRACK 0
8503	027640	000012			12		;CYLINDER 12
8504							
8505	027642	104417			TLOADRK		;LOAD RK REGS
8506	027644	104423			TWAT16		;WAIT FOR INTERRUPT
8507	027646	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8508							
8509	027650	004437	036252		JSR	R4,LRLOAD	;LOAD "L" REGS
8510	027654	000127			WRHEAD		;WRHEAD
8511	027656	177777			-1		; -1 WORDS
8512	027660	072414			OBUFF		;OBUFF IS BUFF ADDRESS
8513	027662	000			.BYTE	0	;SECTOR 0
8514	027663	000			.BYTE	0	;TRACK 0
8515	027664	000012			12		;CYLINDER 12
8516	027666	012737	000040	001616	MOV	#DMD,L.MR1	;SET DIAGNOSTIC-MODE
8517							
8518	027674	104417			TLOADRK		;LOAD RK REGS
8519	027676	004437	036674		JSR	R4,MCLOCK	;CLOCK THROUGH SEEK AND DRIVE CLEAR
8520	027702	001064			1064		
8521							
8522	027704	052762	000200	000026	BIS	#MIND,RKMR1(R2)	;SET INDEX
8523							
8524	027712	004437	036674		JSR	R4,MCLOCK	;CLOCK INDEX
8525	027716	001001			1001		
8526							
8527	027720	042762	000200	000026	BIC	#MIND,RKMR1(R2)	;CLEAR INDEX
8528							
8529	027726	004437	036674		JSR	R4,MCLOCK	;CLOCK CLEAR
8530	027732	001001			1001		
8531							
8532	027734	005062	000026		CLR	RKMR1(R2)	;CLEAR DIAGNOSTIC MODE
8533							
8534	027740	104426			TWAT64		;WAIT FOR INTERRUPT
8535	027742	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8536							
8537	027744	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8538	027746	104004			ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
8539							
8540	027750	004437	036252		JSR	R4,LRLOAD	;LOAD "L" REGS
8541	027754	000101			SELDRV		;SELDRV
8542	027756	000000			0		;0 WORDS
8543	027760	000000			0		;0 IS BUFF ADDRESS
8544	027762	000			.BYTE	0	;SECTOR 0
8545	027763	000			.BYTE	0	;TRACK 0
8546	027764	000012			12		;CYLINDER 12
8547							
8548	027766	005037	001616		CLR	L.MR1	;CLEAR DIAG MODE
8549							
8550	027772	104417			TLOADRK		;LOAD RK REGS

```

8551 027774 104423          TWAT16          ;WAIT FOR INTERRUPT
8552 027776 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
8553
8554 030000 104422          TCHKWE          ;CHECK OPERATION WITH EXPECTED ERROR
8555 030002 040400          UNSERR!DROTERR ;UNSAFE AND DRIVE OFF TRACK
8556 030004 000000          0
8557 030006 000000          0
8558 030010 104004          ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
8559
8560 030012 104416          TSSINIT         ;CLEAR SUBSYSTEM
8561 030014 104003          ERROR 3         ;BAD INIT ERROR
8562
8563 030016 004437 036252          JSR R4,LRLOAD   ;LOAD "L" REGS
8564 030022 000101          SELDRV          ;SELDV
8565 030024 000000          0              ;0 WORDS
8566 030026 000000          0              ;0 IS BUFF ADDRESS
8567 030030 000          .BYTE 0         ;SECTOR 0
8568 030031 000          .BYTE 0         ;TRACK 0
8569 030032 000012          12             ;CYLINDER 12
8570
8571 030034 012737 000001 001616          MOV #1,L.MR1    ;SET MESSAGE SELECT ONE
8572
8573 030042          1$:
8574 030042 104417          TLOADRK         ;LOAD RK REGS
8575 030044 104423          TWAT16          ;WAIT FOR INTERRUPT
8576 030046 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
8577
8578 030050 104421          TCHKOP         ;CHECK OPERATION FOR ANY ERRORS
8579 030052 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8580
8581 030054 032737 000040 001574          BIT #S.HDHM,T.MR2 ;TEST IF HEADS HOME
8582 030062 001767          BEQ 1$
8583
8584 030064 104416          TSSINIT         ;CLEAR SUBSYSTEM
8585 030066 104003          ERROR 3         ;BAD INIT ERROR
8586
8587 030070 005037 001662          CLR INTSET      ;CLEAR INT FLAG
8588 030074 104434          TWAT159        ;WAIT FOR APPROX 160 MS
8589 030076 000240          NOP            ;DON'T CARE ERROR RETURN
8590
8591 030100 104416          TSSINIT         ;CLEAR SUBSYSTEM
8592 030102 104003          ERROR 3         ;BAD INIT ERROR
8593
8594 030104 012762 000100 000000          MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
8595
8596 030112 104437          TWAT8S         ;WAIT FOR SECOND INTERRUPT FROM HDS LOADED
8597 030114 104002          ERROR 2
8598
8599 030116 005037 001616          CLR L.MR1      ;CLEAR MR1
8600
8601 030122 004437 036252          JSR R4,LRLOAD   ;LOAD "L" REGS
8602 030126 000105          CLEAR          ;CLEAR
8603 030130 000000          0              ;0 WORDS
8604 030132 000000          0              ;0 IS BUFF ADDRESS
8605 030134 000          .BYTE 0         ;SECTOR 0
8606 030135 000          .BYTE 0         ;TRACK 0

```



```

8607 030136 000012          12          ;CYLINDER 12
8608
8609 030140 104417          TLOADRK          ;LOAD RK REGS
8610 030142 104423          TWAT16          ;WAIT FOR INTERRUPT
8611 030144 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8612
8613 030146 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8614 030150 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8615
8616 030152 004437 042636          JSR R4,GENCOM   ;BUILD HEADERS
8617 030156 001200
8618
8619 030160 004437 036252          JSR R4,LRLOAD   ;LOAD "L" REGS
8620 030164 000127          WRHEAD          ;WRHEAD
8621 030166 177676          -102            ;-102 WORDS
8622 030170 072414          OBUFF          ;OBUFF IS BUFF ADDRESS
8623 030172 000            .BYTE 0         ;SECTOR 0
8624 030173 000            .BYTE 0         ;TRACK 0
8625 030174 000012          12             ;CYLINDER 12
8626
8627 030176 104417          TLOADRK          ;LOAD RK REGS
8628 030200 104426          TWAT64          ;WAIT FOR INTERRUPT
8629 030202 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8630
8631 030204 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8632 030206 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8633

```

```

*****
*TEST 105 DUMMY TEST FOR PREVIOUS TEST EXIT
* THIS TEST IS PRESENT TO MAKE $SWOBTB TABLE HAVE AN ENTRY
* WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
* IN THE PRECEEDING TEST AND THAT ERROR ABORTS THE TEST.
* IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
* "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
*
* IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
* TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
* FROM THE DRIVE COMING READY AT A LATER TIME.
*****

```

```

8645
8646 030210 000004          ST105: SCOPE
8647 030212 012737 000001 001262      MOV #1,$TIMES ;;DO 1 ITERATION
8648
8649 030220 104416          TSSINIT          ;CLEAR SUBSYSTEM
8650 030222 104003          ERROR 3         ;BAD INIT ERROR
8651
8652 030224 013762 001626 000010      MOV DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
8653 030232 005037 036504          CLR HOLD1
8654 030236 012737 000001 036502      MOV #1,HOLD
8655 030244 053737 001720 036502      BIS DTYP, HOLD
8656 030252 013762 036502 000000      MOV HOLD,RKCS1(R2) ;SELECT THE DRIVE
8657 030260 032762 000200 000012 1$: BIT #DRDY,RKDS(R2) ;TEST IF DRIVE READY
8658 030266 001004          BNE 2$          ;BR IF YES
8659 030270 005237 036504          INC HOLD1       ;ELSE TRY AGAIN
8660 030274 001371          BNE 1$
8661 030276 104002          ERROR 2        ;NO RDY
8662

```

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 160
T105 DUMMY TEST FOR PREVIOUS TEST EXIT

SEQ 0160

```

2$:
8663 030300
8664 030300 104416
8665 030302 104003
8666
8667 030304 004437 042636
8668 030310 001200
8669
8670 030312 004437 036252
8671 030316 000127
8672 030320 177676
8673 030322 072414
8674 030324 000
8675 030325 000
8676 030326 000000
8677
8678 030330 104417
8679 030332 104426
8680 030334 104002
8681
8682 030336 104421
8683 030340 104004
8684 030342 005037 001676

TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

JSR R4,GENCOM ;GENERATE HEADERS FOR CYL 0
1200

JSR R4,LRLoad ;LOAD "L" REGS
WRHEAD ;WRHEAD
-102 ;-102 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0

TLOADRK ;LOAD RK REGS
TWAT64 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
CLR REFMT ;CLEAR REFORMAT SWITCH

```

```

8685 .SBTTL **MULTI-DRIVE OPERATIONS
8686 030346 000004 NEWDRV: SCOPE
8687 030350 012737 000001 001262 MOV #1,$TIMES ;DO ONLY ONCE
8688 030356 032737 000200 001630 BIT #BIT7,DRVBIT ;WERE WE TESTING DRIVE ??
8689 030364 001026 BNE 3$ ;YES-SKIP
8690
8691 030366 005237 001626 1$: INC DRVNUM ;BUMP TO NEXT SEQUENTIAL ADDRESS
8692 030372 006337 001630 ASL DRVBIT ;BUMP DRIVEBIT TO THAT POSITION
8693 030376 033737 001630 001354 BIT DRVBIT,$DEVM ;IS THIS DRIVE TO BE TESTED?
8694 030404 001005 BNE 2$ ;YES-EXIT
8695 030406 032737 000400 001630 BIT #BIT8,DRVBIT ;ALL DRIVES TESTED?
8696 030414 001012 BNE 3$ ;YES-EXIT
8697 030416 000763 BR 1$ ;ELSE CHECK NEXT DRIVE AVAILABLE
8698
8699 030420 112737 000004 001102 2$: MOVB #4,$STSTM ;SET TEST NUMBER FOR REPORTS
8700 030426 013701 001626 MOV DRVNUM,R1
8701 030432 004737 036526 JSR PC,GTYP1 ;GET DRV TYP FOR NEXT DRV TO TST
8702 030436 000137 004742 JMP TSTLUP ;GO TO TEST LOOP TO CHECK THIS DRIVE
8703 030442 005037 001630 3$: CLR DRVBIT ;CLEAR DRIVE BIT
8704 030446 005037 001626 CLR DRVNUM ;CLEAR DRIVE NUMBER
8705
8706 ;*****
8707 ;*TEST 106 RESET ATTENTIONS WITH UNIBUS INIT
8708 ;*
8709 ;* DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
8710 ;* ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.
8711 ;*
8712 ;*****
8713 030452 000004 TST106: SCOPE
8714 030454 012737 000012 001262 MOV #10.,$TIMES ;DO 10. ITERATIONS
8715 030462 005000 CLR R0 ;CLEAR DRIVE POSITION COUNTER
8716 030464 012701 000001 MOV #1,R1 ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
8717 030470 013703 001354 MOV $DEVM,R3 ;GET DEVICE MAP
8718 030474 104416 TSSINIT ;CLEAR SUBSYSTEM
8719 030476 104003 ERROR 3 ;BAD INIT ERROR
8720 030500 030103 1$: BIT R1,R3 ;TEST IF THIS DRIVE AVAILABLE
8721 030502 001006 BNE 2$ ;YES-SKIP TO SEEK
8722 030504 006301 3$: ASL R1 ;SHIFT DRIVE SELECT BIT
8723 030506 005200 INC R0 ;BUMP DRIVE POSITION COUNTER
8724 030510 032701 000400 BIT #BIT8,R1 ;ALL DRIVE POSITIONS CHECKED
8725 030514 001771 BEQ 1$ ;NO-LOOP
8726 030516 000443 BR 4$ ;SKIP TO RESET
8727
8728 030520 010037 001610 2$: MOV R0,L,CS2 ;LOAD DRIVE NUMBER
8729 030524 004737 036512 JSR PC,GTYP0
8730 030530 012737 000113 001600 MOV #RECAL,L,CS1 ;LOAD RECALIBRATE
8731
8732 030536 104417 TLOADRK ;LOAD RK REGS
8733 030540 104423 TWAT16 ;WAIT FOR INTERRUPT
8734 030542 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8735
8736 030544 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
8737 030550 012705 000764 MOV #500.,R5 ;SET COUNT FOR 8 SECONDS
8738 030554 012762 000012 000000 MOV #12,RKCS1(R2) ;RESET INTERRUPT ENABLE
8739 030562 016237 000016 001556 12$: MOV RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8740 030570 113704 001557 MOVB T.ASOF+1,R4 ;ADJUST FOR CHECK OF ATTENTION

```

```

8741 030574 042704 177400      BIC      #177400,R4      ;CLEAR UNUSED BITS
8742 030600 030104              BIT      R1,R4          ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8743 030602 001006      BNE      10$           ;YES - SKIP
8744
8745 030604 104423      TWAT16           ;WAIT FOR 16 MS
8746 030606 000240      NOP              ;DON'T CARE RETURNS
8747 030610 000240      NOP
8748 030612 005305      DEC      R5          ;TATOL WAIT TIME IS 8 SECONDS
8749 030614 001362      BNE      12$         ;CHECK ATTENTION EACH 16 MS
8750 030616 104002      ERROR      2        ;REPORT IF NO ATTENTION IN 8 SEC
8751
8752 030620              10$:
8753 030620 104421      TCHKOP           ;CHECK OPERATION FOR ANY ERRORS
8754 030622 104004      ERROR      4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8755
8756 030624 000727      BR           ;LOOP FOR NEXT DRIVE
8757 030626 000005      4$: RESET          ;UNIBUS RESET
8758 030630 004737 045524      JSR      PC,$TKINT ;RESET KEYBOARD INTERRUPT ENABLE
8759
8760 030634 012701 000031      MOV      #25.,R1     ;DO A SHORT DELAY
8761 030640 005301      DEC      R1
8762 030642 001376      BNE      5$
8763 030644 004737 035120      JSR      PC,OPTTST  ;SET UP OPTIONS
8764
8765 030650 104420      TGETRK          ;GET RK611 REGS
8766
8767 030652 105737 001557      TSTB      T.ASOF+1  ;ALL ATTENTION RESET?
8768 030656 001407      BEQ      6$        ;YES-SKIP
8769
8770 030660 012737 057172 001470      MOV      #EMDA,EM12N ;"DRIVE ATT NOT RESET RESULT OF
8771 030666 012737 057316 061454      MOV      #EMUR,DF011A ;UNIBUS RESET"
8772 030674 104012      ERROR      12
8773
8774 030676      6$:
8775      ;*****
8776      ;*TEST 107      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
8777      ;*
8778      ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
8779      ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
8780      ;*      RESET.
8781      ;*
8782      ;*****
8783 030676 000004      †ST107: SCOPE
8784 030700 012737 000012 001262      MOV      #10.,$TIMES ;DO 10. ITERATIONS
8785 030706 005000      CLR      R0          ;CLEAR DRIVE POSITION COUNTER
8786 030710 012701 000001      MOV      #1,R1       ;PRESET TO TEST POSITION 0
8787 030714 013703 001354      MOV      $DEVN,R3    ;CUT DEVICE MAP
8788 030720 104416      TSSINIT          ;CLEAR SUBSYSTEM
8789 030722 104003      ERROR      3        ;BAD INIT ERROR
8790 030724 030103      1$: BIT      R1,R3    ;THIS DRIVE AVAILABLE?
8791 030726 001006      BNE      2$        ;YES-SKIP TO SEEK
8792 030730 006301      3$: ASL      R1          ;SHIFT TO NEXT DRIVE POSITION
8793 030732 005200      INC      R0          ;DUMP POSITION COUNTER
8794 030734 032701 000400      BIT      #BIT8,R1   ;ALL POSITIONS CHECKED
8795 030740 001771      BEQ      1$        ;NO-LOOP
8796 030742 000443      BR       4$        ;YES-SKIP TO CLEAR

```

```

8797
8798 030744 010037 001610      2$:  MOV      RD,L,CS2      ;LOAD DRIVE NUMBER
8799 030750 004737 036512      JSR      PC,GTYPC
8800 030754 012737 000113 001600  MOV      #RECAL,L,CS1    ;LOAD RECALIBRATE
8801 030762 104417      TLOADRK ;LOAD RK REGS
8802 030764 104423      TWAT16  ;WAIT FOR INTERRUPT
8803 030766 104002      ERROR  2      ;TO SLOW/NOT COMPLETE ERROR
8804
8805 030770 005037 001662      CLR      INTSET        ;CLEAR INT FLAG
8806 030774 012705 000764      MOV      #500.,R5      ;SET COUNT FOR 8 SECONDS
8807 031000 012762 000012 000000  MOV      #12,RKCS1(R2)  ;RESET INTERRUPT ENABLE
8808 031006 016237 000016 001556 12$:  MOV      RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8809 031014 113704 001557      MOV      T.ASOF+1,R4    ;ADJUST FOR CHECK OF ATTENTION
8810 031020 042704 177400      BIC      #177400,R4    ;CLEAR UNUSED BITS
8811 031024 030104      BIT      R1,R4         ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8812 031026 001006      BNE      10$          ;YES - SKIP
8813
8814 031030 104423      TWAT16  ;WAIT FOR 16 MS
8815 031032 000240      NOP      ;DON'T CARE RETURNS
8816 031034 000240      NOP
8817 031036 005305      DEC      R5           ;TATOL WAIT TIME IS 8 SECONDS
8818 031040 001362      BNE      12$          ;CHECK ATTENTION EACH 16 MS
8819 031042 104002      ERROR  2      ;REPORT IF NO ATTENTION IN 8 SEC
8820
8821 031044      10$:
8822 031044 104421      TCHKOP  ;CHECK OPERATION FOR ANY ERRORS
8823 031046 104004      ERROR  4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8824
8825 031050 000727      BR       3$           ;LOOP FOR NEXT DRIVE
8826
8827 031052 052762 000040 000010 4$:  BIS      #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
8828 031060 012701 000031      MOV      #25.,R1      ;DO A SHORT DELAY
8829 031064 005301      DEC      R1
8830 031066 001376      BNE      5$
8831
8832 031070 104420      TGETRK  ;GET RK611 REGS
8833
8834 031072 105737 001557      TSTB    T.ASOF+1      ;TEST ALL ATTENTION RESET
8835 031076 001407      BEQ     6$           ;YES-SKIP
8836
8837 031100 012737 057172 001470  MOV      #EMDA,EM12N   ;"DRIVE ATT NOT RESET AS RESULT OF
8838 031106 012737 057406 061454  MOV      #EMSCLR,DF011A ;SUBSYSTEM CLEAR"
8839 031114 104012      ERROR  12
8840
8841 031116      6$:
8842 *****
8843 *TEST 110      SVAL AND ATTENTION FROM OTHER DRIVE
8844 *
8845 *      DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT
8846 *      ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID
8847 *      IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
8848 *      MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
8849 *      CHANGE REMAINS RESET.
8850 *
8851 *      REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
8852 *

```

```

8853      ;*      NOTE:  THIS TEST WILL ONLY BE DONE IF AT LEAST
8854      ;*      TWO DRIVES ARE AVAILABLE.
8855      ;*
8856      ;*****
8857      †ST110: SCOPE
8858      031116 000004      000012 001262      MOV      #10, $TIMES      ;; DO 10. ITERATIONS
8859      031120 012737      001354      MOV      $DEVN, -(SP)      ;; PUT DEVICE MAP ON STACK
8860      031132 004437      036620      JSR      R4, BITCNT      ;; COUNT NUMBER OF BITS (# OF DRIVES)
8861      031136 022627      000001      CMP      (SP)+, #1      ;; COMPARE TO 1
8862      031142 101007      BHI      2$      ;; SKIP IF MORE THAN 1
8863      031144 005737      001304      TST      $PASS      ;; CHECK IF PASS 0
8864      031150 001002      BNE      1$      ;; NO-SKIP
8865
8866      031152 104401      053226      TYPE      OPR018      ;; "OVERLAPPED OPERATION BYPASSED"
8867      031156 000137      032422      1$:      JMP      $EOP      ;; GET OUT.
8868
8869      031162 012737      177777      001224      2$:      MOV      #-1, $TMP1      ;; SET LOOP CONTROL FLAG
8870      031170 013705      001354      MOV      $DEVN, R5      ;; GET DEVICE MAP
8871      031174 005000      CLR      R0      ;; CLEAR FOR DRIVE #A
8872      031176 005001      CLR      R1      ;; CLEAR FOR DRIVE #B
8873      031200 012703      000001      MOV      #1, R3      ;; SET DRIVE POSITION A
8874      031204 012704      000001      MOV      #1, R4      ;; SET DRIVE POSITION B
8875      031210 012737      031220      001110      MOV      #3$, $LPERR      ;; SET LOCAL LOOP ON ERROR
8876      031216 000503      BR       11$      ;; GO SET UP POINTERS
8877
8878      031220      3$:
8879      031220 104416      TSSINIT      ;; CLEAR SUBSYSTEM
8880      031222 104003      ERROR      3      ;; BAD INIT ERROR
8881
8882      031224 010037      001610      MOV      R0, L.CS2      ;; LOAD DRIVE A
8883      031230 010037      001626      MOV      R0, DRVNUM      ;; LOAD FOR REPORT
8884      031234 004737      036512      JSR      PC, GTYP0
8885      031240 012737      000113      001600      MOV      #RECAL, L.CS1      ;; LOAD RECAL
8886
8887      031246 104417      TLOADRK      ;; LOAD RK REGS
8888      031250 104423      TWAT16      ;; WAIT FOR INTERRUPT
8889      031252 104002      ERROR      2      ;; TO SLOW/NOT COMPLETE ERROR
8890      031254 104421      TCHKOP      ;; CHECK OPERATION FOR ANY ERRORS
8891      031256 104004      ERROR      4 ;OR 5, 6, 7      ;; REPORT ALL ERRORS
8892
8893      031260 005037      001662      CLR      INTSET      ;; CLEAR INT FLAG
8894
8895      031264 010137      001610      MOV      R1, L.CS2      ;; LOAD DRIVE B
8896      031270 010137      001626      MOV      R1, DRVNUM      ;; LOAD FOR REPORT
8897      031274 004737      036526      JSR      PC, GTYP1
8898      031300 012737      000101      001600      MOV      #SELDRV, L.CS1      ;; LOAD DRIVE SELECT
8899
8900      031306 104417      TLOADRK      ;; LOAD RK REGS
8901      031310 104423      TWAT16      ;; WAIT FOR INTERRUPT
8902      031312 104002      ERROR      2      ;; TO SLOW/NOT COMPLETE ERROR
8903
8904      031314 104421      TCHKOP      ;; CHECK OPERATION FOR ANY ERRORS
8905      031316 104004      ERROR      4 ;OR 5, 6, 7      ;; REPORT ALL ERRORS
8906
8907      031320 032737      100000      001552      BIT      #SVAL, T.DS      ;; CHECK IF STATUS VALID SET
8908      031326 001007      BNE      4$      ;; YES - SKIP

```

```

8909 031330 012737 057006 001460      MOV      #EMSVAL,EM11N      ;"STATUS VALID NOT SET RESULT OF
8910 031336 012737 050536 061454      MOV      #OPER00,DF011A    ;DRIVE SELECT"
8911 031344 104011                                ERROR    11
8912
8913 031346 005037 001662      4$:     CLR      INTSET      ;CLEAR INT FLAG
8914 031352 104436                                TWAT2S   ;WAIT FOR SEEK COMPLETE INTERRUPT
8915 031354 000401                                BR       44$              ;NONE RECEIVED - SKIP
8916 031356 000406                                BR       55$              ;RECEIVED - SKIP
8917
8918 031360 010037 001626      44$:    MOV      R0,DRVNUM1        ;SET DRIVE FOR REPORT
8919 031364 012737 061043 001372      MOV      #DH017,DH2N      ;"COMMAND - SELECT AFTER RECAL"
8920 031372 104002                                ERROR    2
8921
8922 031374 104420                                TGETRK   ;GET RK REGS
8923 031376 032737 100000 001552      55$:    BIT      #SVAL,T.DS      ;TEST IF SVAL STILL SET
8924 031404 001007                                BNE     5$                ;YES - SKIP
8925
8926 031406 012737 057006 001510      MOV      #EMSVAL,EM14N      ;"STATUS VALID RESET RESULT OF
8927 031414 012737 061066 061454      MOV      #DH018,DF011A    ;RECAL COMPLETE ATTENTION AFTER SEL"
8928 031422 104014                                ERROR    14
8929
8930 031424 104415      5$:     SCOP1              ;LOCAL LOOP TO 3$
8931
8932      ;
8933      ; THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8934      ; OF DRIVES AVAILABLE. THE FIRST PASS OF THE PROGRAM WILL USE THE
8935      ; LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8936      ; B. THE SECOND PASS SWAPS DRIVE A & B. THE THIRD PASS USES
8937      ; THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8938      ; AS A. THE FORTH PASS SWAPS A & B AGAIN. THIS CONTINUES
8939      ; UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8940      ; DRIVE.
8941      ;
8942      ; THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8943      ; THIRD HIGHEST AS B. THEY ARE SWAPPED ON THE NEXT PASS.
8944      ;
8945      ; THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8946      ; CHECKED.
8947      ;
8947 031426 005237 001224      11$:    INC      $TMP1          ;INCREMENT PASS CONTROL
8948 031432 001024                                BNE     16$              ;SKIP IF NOT ZERO
8949                                ;(IT WILL BE ZERO ON THE 1ST PASS)
8950
8951 031434 030305      12$:    BIT      R3,R5          ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8952 031436 001006                                BNE     13$              ;YES-SKIP
8953
8954 031440 005200      22$:    INC      R0              ;BUMP R0 (DRIVE A)
8955 031442 006303                                ASL     R3                ;SHIFT DRIVE SELECT BIT ONE POSITION
8956 031444 032703 000400      BIT      #BIT8,R3        ;IF BIT 8 IS SET, ALL DRIVES HAVE
8957 031450 001771                                BEQ     12$              ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8958 031452 000464                                BR      50$              ;DONE-EXIT
8959
8960 031454 010001      13$:    MOV      R0,R1          ;SET DRIVE B TO THE SAME AS A
8961 031456 010304                                MOV     R3,R4
8962 031460 005201      14$:    INC      R1              ;BUMP R1 (DRIVE B)
8963 031462 006304                                ASL     R4                ;SHIFT SELECTOR BIT ONE POSITION
8964 031464 030405                                BIT     R4,R5              ;IS THIS DRIVE AVAIL?

```

K13

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 166
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0166

```

8965 031466 001004          BNE      15$          ;YES-SKIP
8966 031470 032704 000400  BIT      #BIT8,R4    ;WERE ALL POSITIONS CHECKED?
8967 031474 001771          BEQ      14$          ;NO-LOOP
8968 031476 000452          BR       50$          ;DONE-EXIT
8969
8970 031500 000137 031220    15$:    JMP      3$          ;GO DO THE TEST ON THE DRIVE A & B
8971                                ;CONTAINED IN R0 & R1
8972 031504 032737 000001 001224  16$:    BIT      #BIT0,$TMP1 ;IS PASS FLAGS ODD?
8973 031512 001410          BEQ      17$          ;NO-SKIP
8974
8975 031514 010046          MOV      R0,-(SP)    ;
8976 031516 010346          MOV      R3,-(SP)    ;SWAP R0 & R1, R3 & R4
8977 031520 010403          MOV      R4,R3      ;TO EXCHANGE DRIVE A & B
8978 031522 010100          MOV      R1,R0
8979 031524 012604          MOV      (SP)+,R4
8980 031526 012601          MOV      (SP)+,R1
8981 031530 000137 031220    JMP      3$          ;REPEAT TEST ON THIS COMBO.
8982
8983 031534 032737 000002 001224  17$:    BIT      #BIT1,$TMP1 ;TEST IF PASS FLAGS AT HALF MODULE 4?
8984 031542 001410          BEQ      19$          ;NO-SKIP TO BUMP DRIVE B
8985 031544 005200          18$:    INC      R0          ;BUMP DRIVE A
8986 031546 006303          ASL      R3          ;SHIFT DRIVE SELECT BIT
8987 031550 030305          BIT      R3,R5      ;AVAILABLE?
8988 031552 001014          BNE      20$          ;YES-SKIP
8989 031554 032703 000400  BIT      #BIT8,R3    ;ALL CHECKED?
8990 031560 001771          BEQ      18$          ;NO-SKIP
8991 031562 000412          BR       21$          ;GO TO NEXT PASS
8992
8993 031564 005201          19$:    INC      R1          ;BUMP DRIVE B
8994 031566 006304          ASL      R4          ;SHIFT DRIVE SELECT BIT
8995 031570 030405          BIT      R4,R5      ;AVAILABLE?
8996 031572 001004          BNE      20$          ;YES-SKIP
8997 031574 032704 000400  BIT      #BIT8,R4    ;ALL CHECKED?
8998 031600 001771          BEQ      19$          ;NO-LOOP
8999 031602 000404          BR       23$          ;YES-SKIP TO NEXT PASS
9000
9001 031604 000137 031220    20$:    JMP      3$          ;GO TEST THIS COMBO
9002
9003 031610 010100          21$:    MOV      R1,R0      ;SET DRIVE 0 TO LOW POSITION THIS PASS
9004 031612 010403          MOV      R4,R3      ;SET SELECT BITS TO AGREE
9005 031614 005037 001224  23$:    CLR      $TMP1      ;CLEAR PASS FLAGS
9006 031620 000137 031440  JMP      22$          ;GO SET UP A & B
9007 031624
9008 50$:
9009 *****
9010 *TEST 111 OVERLAPPED OPERATIONS
9011 *
9012 * DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A
9013 * SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE
9014 * DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
9015 * AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
9016 * BE SET AND DRIVE A HAS ATTENTION SET.
9017 *
9018 * REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
9019 *
9020 * NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE
          TEST WILL NOT BE DONE.

```



```

9021 ;*
9022 ;*****
9023 031624 000004          †ST111: SCOPE
9024 031626 012737 000005 001262  MOV      #5.,$TIMES      ;;DO 5. ITERATIONS
9025
9026 031634 012737 177777 001224 2$:      MOV      #-1,$TMP1      ;SET LOOP CONTROL FLAG
9027 031642 013705 001354          MOV      $DEVM,R5      ;GET DEVICE MAP
9028 031646 005000          CLR      R0            ;CLEAR FOR DRIVE #A
9029 031650 005001          CLR      R1            ;CLEAR FOR DRIVE #B
9030 031652 012703 000001          MOV      #1,R3        ;SET DRIVE POSITION A
9031 031656 012704 000001          MOV      #1,R4        ;SET DRIVE POSITION B
9032 031662 012737 031672 001110  MOV      #3,$$LPERR   ;SET LOCAL LOOP ON ERROR
9033 031670 000555          BR       11$          ;GO SET UP POINTERS
9034 031672
9035 031672 104416          3$:      TSSINIT          ;CLEAR SUBSYSTEM
9036 031674 104003          ERROR   3            ;BAD INIT ERROR
9037
9038 031676 010037 001626          MOV      R0,DRVNUM    ;STORE DRIVE FOR REPORT
9039 031702 010037 001610          MOV      R0,L.CS2    ;SETUP DRIVE A TO RECAL
9040 031706 004737 036512          JSR     PC,GTYPO
9041 031712 012737 000113 001600  MOV      #RECAL,L.CS1
9042
9043 031720 104417          TLOADRK          ;LOAD RK REGS
9044 031722 104423          TWAT16          ;WAIT FOR INTERRUPT
9045 031724 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9046 031726 005037 001662  CLR      INTSET     ;CLEAR INTERRUPT FLAG
9047
9048 031732 104437          TWAT8S          ;WAIT FOR SECOND INTERRUPT
9049 031734 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9050
9051 031736 012737 000105 001600  MOV      #CLEAR,L.CS1 ;SET UP TO CLEAR DRIVE
9052 031744 104417          TLOADRK          ;LOAD RK REGS
9053 031746 104423          TWAT16          ;WAIT FOR INTERRUPT
9054 031750 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9055 031752
9056 031752 104421          4$:      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
9057 031754 104004          ERROR   4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
9058 031756 032737 040000 001540  BIT      #DI,T.CS1    ;TEST IF DI STILL SET
9059 031764 001372          BNE     4$        ;YES - LOOP
9060 031766 010137 001626          MOV      R1,DRVNUM    ;STORE DRIVE FOR REPORT
9061 031772 010137 001610          MOV      R1,L.CS2    ;SETUP DRIVE B TO RECAL
9062 031776 004737 036526          JSR     PC,GTYPI
9063 032002 012737 000113 001600  MOV      #RECAL,L.CS1
9064
9065 032010 104417          TLOADRK          ;LOAD RK REGS
9066 032012 104423          TWAT16          ;WAIT FOR INTERRUPT
9067 032014 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9068 032016 005037 001662  CLR      INTSET     ;CLEAR INTERRUPT FLAG
9069 032022 104437          TWAT8S          ;WAIT FOR SECOND INTERRUPT
9070 032024 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9071 032026 012737 000105 001600  MOV      #CLEAR,L.CS1 ;SET UP DRIVE CLEAR
9072 032034 104417          TLOADRK          ;LOAD RK REGS
9073 032036 104423          TWAT16          ;WAIT FOR INTERRUPT
9074 032040 104002          ERROR   2        ;TO SLOW/NOT COMPLETE ERROR
9075 032042
9076 032042 104421          5$:      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS

```

M13

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 168
T111 OVERLAPPED OPERATIONS

SEQ 0168

9077	032044	104004				ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
9078	032046	032737	040000	001540		BIT	#DI,T.CS1	;TEST IF DI STILL SET
9079	032054	001372				BNE	5\$;YES - LOOP
9080								
9081	032056	010037	001626			MOV	RO,DRVNUM	;STORE DRIVE FOR REPORT
9082	032062	010037	001610			MOV	RO,L.CS2	;SETUP DRIVE A TO SEEK
9083	032066	004737	036512			JSR	PC,GTYP0	
9084	032072	012737	000001	001614		MOV	#1,L.DCYL	;TO CYL 1
9085	032100	012737	000117	001600		MOV	#SEEK,L.CS1	
9086								
9087	032106	104417				TLOADRK		;LOAD RK REGS
9088	032110	104423				TWAT16		;WAIT FOR INTERRUPT
9089	032112	104002				ERROR	2	;TO SLOW/NOT COMPLETE ERROR
9090								
9091	032114	010137	001626			MOV	R1,DRVNUM	;STORE DRIVE FOR REPORT
9092	032120	010137	001610			MOV	R1,L.CS2	;SETUP DRIVE B TO WRITE DATA
9093	032124	004737	036526			JSR	PC,GTYP1	
9094	032130	012737	000100	001614		MOV	#100,L.DCYL	;AT CYL 100
9095	032136	012737	177400	001602		MOV	#-400,L.WC	;400 WORDS
9096	032144	012737	072414	001604		MOV	#OBUFF,L.BA	
9097	032152	012737	000123	001600		MOV	#WRDATA,L.CS1	
9098								
9099	032160	104417				TLOADRK		;LOAD RK REGS-DO WRITE
9100								
9101	032162	104427				TWAT80		;WAIT FOR INTERRUPT
9102	032164	104002				ERROR	2	;TO SLOW/NOT COMPLETE ERROR
9103								
9104	032166	104421				TCHKOP		;CHECK OPERATION FOR ANY ERRORS
9105	032170	104004				ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
9106								
9107	032172	010037	001626			MOV	RO,DRVNUM	;STORE DRIVE FOR REPORT
9108	032176	130337	001557			BITB	R3,T.ASOF+1	;CHECK IF DRIVE ATTENTION IS DRIVE A
9109	032202	001007				BNE	10\$;YES-SKIP
9110	032204	012737	057172	001460		MOV	#EMDA,EM11N	;DRIVE ATTENTION NOT SET RESULT OF
9111	032212	012737	050650	061454		MOV	#EMSK,DF011A	;SEEK"
9112	032220	104011				ERROR	11	
9113								
9114	032222	104415			10\$:	SCOPI		;LOCAL LOOP TO 3\$
9115								
9116								
9117								
9118								
9119								
9120								
9121								
9122								
9123								
9124								
9125								
9126								
9127								
9128								
9129								
9130								
9131	032224	005237	001224		11\$:	INC	\$TMP1	;INCREMENT PASS CONTROL
9132	032230	001024				BNE	16\$;SKIP IF NOT ZERO

THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION OF DRIVES AVAILABLE. THE FIRST PASS OF THE PROGRAM WILL USE THE LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS B. THE SECOND PASS SWAPS DRIVE A & B. THE THIRD PASS USES THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE AS A. THE FORTH PASS SWAPS A & B AGAIN. THIS CONTINUES UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER DRIVE.

THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE THIRD HIGHEST AS B. THEY ARE SWAPPED ON THE NEXT PASS.

THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE CHECKED.

N13

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 169
T111 OVERLAPPED OPERATIONS

SEQ 0169

```

9133                                     ;(IT WILL BE ZERO ON THE 1ST PASS)
9134
9135 032232 030305          12$:  BIT    R3,R5          ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
9136 032234 001006          BNE    13$              ;YES-SKIP
9137
9138 032236 005200          22$:  INC    R0              ;BUMP R0 (DRIVE A)
9139 032240 006303          ASL    R3              ;SHIFT DRIVE SELECT BIT ONE POSITION
9140 032242 032703 000400  BIT    #BIT8,R3          ;IF BIT 8 IS SET, ALL DRIVES HAVE
9141 032246 001771          BEQ    12$              ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
9142 032250 000464          BR     50$              ;DONE-EXIT
9143
9144 032252 010001          13$:  MOV    R0,R1          ;SET DRIVE B TO THE SAME AS A
9145 032254 010304          MOV    R3,R4
9146 032256 005201          14$:  INC    R1              ;BUMP R1 (DRIVE B)
9147 032260 006304          ASL    R4              ;SHIFT SELECTOR BIT ONE POSITION
9148 032262 030405          BIT    R4,R5          ;IS THIS DRIVE AVAIL?
9149 032264 001004          BNE    15$              ;YES-SKIP
9150 032266 032704 000400  BIT    #BIT8,R4          ;WERE ALL POSITIONS CHECKED?
9151 032272 001771          BEQ    14$              ;NO-LOOP
9152 032274 000452          BR     50$              ;DONE-EXIT
9153
9154 032276 000137 031672  15$:  JMP    3$              ;GO DO THE TEST ON THE DRIVE A & B
9155
9156 032302 032737 000001 001224 16$:  BIT    #BIT0,$TMP1          ;CONTAINED IN R0 & R1
9157 032310 001410          BEQ    17$              ;IS PASS FLAGS ODD?
9158
9159 032312 010046          MOV    R0,-(SP)          ;
9160 032314 010346          MOV    R3,-(SP)          ;SWAP R0 & R1, R3 & R4
9161 032316 010403          MOV    R4,R3            ;TO EXCHANGE DRIVE A & B
9162 032320 010100          MOV    R1,R0
9163 032322 012604          MOV    (SP)+,R4
9164 032324 012601          MOV    (SP)+,R1
9165 032326 000137 031672  JMP    3$              ;REPEAT TEST ON THIS COMBO.
9166
9167 032332 032737 000002 001224 17$:  BIT    #BIT1,$TMP1          ;TEST IF PASS FLAGS AT HALF MODULE 4?
9168 032340 001410          BEQ    19$              ;NO-SKIP TO BUMP DRIVE B
9169 032342 005200          18$:  INC    R0              ;BUMP DRIVE A
9170 032344 006303          ASL    R3              ;SHIFT DRIVE SELECT BIT
9171 032346 030305          BIT    R3,R5          ;AVAILABLE?
9172 032350 001014          BNE    20$              ;YES-SKIP
9173 032352 032703 000400  BIT    #BIT8,R3          ;ALL CHECKED?
9174 032356 001771          BEQ    18$              ;NO-SKIP
9175 032360 000412          BR     21$              ;GO TO NEXT PASS
9176
9177 032362 005201          19$:  INC    R1              ;BUMP DRIVE B
9178 032364 006304          ASL    R4              ;SHIFT DRIVE SELECT BIT
9179 032366 030405          BIT    R4,R5          ;AVAILABLE?
9180 032370 001004          BNE    20$              ;YES-SKIP
9181 032372 032704 000400  BIT    #BIT8,R4          ;ALL CHECKED?
9182 032376 001771          BEQ    19$              ;NO-LOOP
9183 032400 000404          BR     23$              ;YES-SKIP TO NEXT PASS
9184
9185 032402 000137 031672  20$:  JMP    3$              ;GO TEST THIS COMBO
9186
9187 032406 010100          21$:  MOV    R1,R0          ;SET DRIVE D TO LOW POSITION THIS PASS
9188 032410 010403          MOV    R4,R3          ;SET SELECT BITS TO AGREE

```

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 170
T111 OVERLAPPED OPERATIONS

SEQ 0170

9189 032412 005037 001224
9190 032416 000137 032236
9191 032422

23\$: CLR \$TMP1 ;CLEAR PASS FLAGS
JMP 22\$;GO SET UP A & B
50\$:

```

9192 ;ADD WAITING LOOP FOR APT OPERATION ON JAN-9-78 REV 007
9193 .SBTTL END OF PASS ROUTINE
9194
9195 ;*****
9196 ;*INCREMENT THE PASS NUMBER ($PASS)
9197 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
9198 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
9199 ;*IF THERES A MONITOR GO TO IT
9200 ;*IF THERE ISN'T JUMP TO TSTAPT
9201
9202 $EOP:
9203 032422 000004 SCOPE
9204 032424 005037 001102 CLR $TSTNM ;; ZERO THE TEST NUMBER
9205 032430 005037 001262 CLR $TIMES ;; ZERO THE NUMBER OF ITERATIONS
9206 032434 005237 001304 INC $PASS ;; INCREMENT THE PASS NUMBER
9207 032440 042737 100000 001304 BIC #100000,$PASS ;; DON'T ALLOW A NEG. NUMBER
9208 032446 005327 DEC (PC)+ ;; LOOP?
9209 032450 000001 $EOPCT: .WORD 1
9210 032452 003063 BGT $DOAGN ;; YES
9211 032454 012737 MOV (PC)+,a(PC)+ ;; RESTORE COUNTER
9212 032456 000001 $ENDCT: .WORD 1
9213 032460 032450 $EOPCT
9214 032462 104401 032470 TYPE 65$ ;; TYPE ASCIZ STRING
9215 032466 000407 BR 64$ ;; GET OVER THE ASCIZ
9216
9217 032506 013746 001304 ;; 65$: .ASCIZ <12><15>/END PASS #/
9218 032506 013746 001304 64$: MOV $PASS,-(SP) ;; SAVE $PASS FOR TYPEOUT
9219
9220 032512 104405 TYPDS ;; TYPE PASS NUMBER
9221 032514 104401 032522 TYPE 67$ ;; GO TYPE--DECIMAL ASCII WITH SIGN
9222 032520 000421 BR 66$ ;; TYPE ASCIZ STRING
9223
9224 032564 013746 001112 ;; 67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
9225 032564 013746 001112 66$: MOV $ERTTL,-(SP) ;; SAVE $ERTTL FOR TYPEOUT
9226
9227 032570 104405 TYPDS ;; TOTAL NUMBER OF ERRORS
9228 032572 104401 001273 TYPE 67$ ;; GO TYPE--DECIMAL ASCII WITH SIGN
9229 032576 005037 001112 CLR $ERTTL ;; TYPE CARRIAGE RETURN, LINE FEED
9230 032602 013700 000042 $GET42: MOV a#42,RO ;; CLEAR ERROR TOTAL
9231 032606 001405 BEQ $DOAGN ;; GET MONITOR ADDRESS
9232 032610 000005 RESET ;; BRANCH IF NO MONITOR
9233 032612 004710 $SENDAD: JSR PC,(RO) ;; CLEAR THE WORLD
9234 032614 000240 NOP ;; GO TO MONITOR
9235 032616 000240 NOP ;; SAVE ROOM
9236 032620 000240 NOP ;; FOR
9237 032622 $DOAGN: NOP ;; ACT11
9238 032622 000137 JMP a(PC)+ ;; RETURN
9239 032624 032632 $RTNAD: .WORD TSTAPT
9240 032626 377 377 000 $ENULL: .BYTE -1,-1,0 ;; NULL CHARACTER STRING
9241
9242 032632 122737 000001 001316 TSTAPT: CMPB #APTENV,$ENV ;; RUNNING UNDER APT MODE ?
9243 032640 001007 BNE 2$ ;; BRANCH IF NOT
9244 032642 023727 001304 000002 CMP $PASS,#2 ;; TWO PASS DONE ?
9245 032650 103403 BLO 2$ ;; BRANCH IF NOT
9246 032652 005237 001102 1$: INC $TSTNM ;; INCREMENT TEST NUMBER
9247 032656 000775 BR 1$ ;; WAITING LOOP FOR APT DUMP THE NEXT PROG

```

```

9248 032660 000137 003204
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265 032664 010046
9266 032666 010146
9267 032670 010246
9268 032672 010346
9269 032674 013746 000004
9270 032700 013746 000006
9271 032704 010600
9272
9273 032706 104400
9274 032710 012637 000006
9275 032714 012701 003776
9276 032720 105727
9277 032722 000200
9278 032724 100062
9279 032726 012737 033064 000004
9280 032734 005737 177572
9281 032740 052737 100000 032722
9282 032746 005046
9283 032750 012702 172340
9284 032754 012703 000010
9285 032760 012762 077406 177740 1$:
9286 032766 011622
9287 032770 062716 000200
9288 032774 077307
9289 032776 012742 177600
9290 033002 005042
9291 033004 012737 033022 000004
9292 033012 012737 000020 172516
9293 033020 000401
9294 033022 022626
9295 033024 005237 177572
9296 033030 012737 033054 000004
9297 033036 005737 143776
9298 033042 062712 000040
9299 033046 023712 172356
9300 033052 101371
9301 033054 011202
9302 033056 005037 177572
9303 033062 000421

```

```

2$: JMP TST1 ;REPRT THE SAME TEST IF NOT UNDER APT
.SBTTL ROUTINE TO SIZE MEMORY
*****
*CALL:
* JSR PC,$SIZE
* RETURN
*$LSTAD WILL CONTAIN:
* WITH KT11 OPTION -- LAST VIRTUAL ADDRESS OF THE LAST BANK
* WITHOUT KT11 OPTION -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
*$LSTBK WILL CONTAIN THE LAST BANK AS A SAF
*$KT11 IS THE MEMORY MANAGEMENT KEY
*$BIT07 = 0 DON'T USE MEMORY MANAGEMENT
* MUST BE SETUP BEFORE THE CALL
*$BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
* DETERMINED BY ROUTINE

$SIZE: MOV R0,-(SP) ;;SAVE R0 ON THE STACK
MOV R1,-(SP) ;;SAVE R1 ON THE STACK
MOV R2,-(SP) ;;SAVE R2 ON THE STACK
MOV R3,-(SP) ;;SAVE R3 ON THE STACK
MOV @#ERRVEC,-(SP) ;;SAVE PRESENT ERROR VECTOR PS & PC
MOV @#ERRVEC+2,-(SP)
MOV SP,R0 ;;SAVE THE STACK POINTER
;;SET THE ERRVEC PS TO THE PRESENT PS
TRAP ;;PUSH OLD PSW AND PC ON STACK
MOV (SP)+,@#ERRVEC+2 ;;SAVE THE PSW IN @#ERRVEC+2
MOV #3776,R1 ;;SETUP ADDRESS
MOV (PC)+ ;;USE MEMORY MANAGEMENT?
$KT11: .WORD 200 ;;SET TO USE MEMORY MANAGEMENT
BPL SCORE ;;BR IF NO
MOV #S$KTNEX,@#ERRVEC ;;SET FOR TIMEOUT
TST @#SR0 ;;KT11 ARE YOU THERE?
BIS #100000,$KT11 ;;YES--SET KT11 KEY
CLR -(SP) ;;INITIALIZE FOR "PAR" LOADING
MOV #KIPAR0,R2 ;;ADDRESS OF FIRST "PAR"
MOV #1D8,R3 ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
1$: MOV #77406,-40(R2) ;;PDR = 4K, UP, READ/WRITE
MOV (SP),(R2)+ ;;LOAD "PAR"
ADD #200,(SP) ;;UPDATE FOR NEXT "PAR"
SOB R3,1$ ;;LOOP UNTIL ALL EIGHT ARE LOADED
MOV #177600,-(R2) ;;SETUP KIPAR7 FOR I/O
CLR -(R2) ;;SETUP KIPAR6 FOR TESTING
MOV #2$,@#ERRVEC ;;CATCH TIMEOUT IF NO SR3
MOV #20,@#SR3 ;;ENABLE 22 BIT MODE
BR 3$ ;;THIS PDP-11 HAS A SR3 REGISTER
2$: CMP (SP)+,(SP)+ ;;CLEAN OFF THE STACK--NO SR3
3$: INC @#SR0 ;;TURN ON MEMORY MANAGEMENT
MOV #S$KTOUT,@#ERRVEC ;;SET FOR TIME OUT
4$: TST @#143776 ;;TRAP ON NON-EX-MEM
ADD #40,(R2) ;;MAKE A 1K STEP
CMP @#KIPAR7,(R2) ;;LAST ONE?
BHI 4$ ;;NO--TRY IT
$KTOUT: MOV (R2),R2 ;;GET LAST BANK+1
CLR @#SR0 ;;TURN OFF MEMORY MANAGEMENT
BR $SIZEX

```

```

9304 033064 042737 100000 032722 $KTNEX: BIC #100000,$KT11 ;;KT11 NON-EXISTENT
9305 033072 012737 033122 000004 $SCORE: MOV #$SCROUT,$ERRVEC ;;SET FOR TIMEOUT
9306 033100 005002 CLR R2 ;;SET UP BANK
9307 033102 062701 004000 1$: ADD #4000,R1 ;;INCREMENT BY 1K
9308 033106 062702 000040 ADD #40,R2 ;;1K STEP
9309 033112 005711 TST (R1) ;;TRAP ON TIME OUT
9310 033114 022701 177776 CMP #177776,R1 ;;LAST ONE
9311 033120 001370 BNE 1$ ;;NO--TRY AGAIN
9312 033122 162701 004000 $SCROUT: SUB #4000,R1
9313 033126 162702 000040 $$SIZE: SUB #40,R2 ;;DROP BACK
9314 033132 010006 MOV RO,SP ;;RESTORE THE STACK
9315 033134 012637 000006 MOV (SP)+,$ERRVEC+2 ;;RESTORE ERROR VECTOR
9316 033140 012637 000004 MOV (SP)+,$ERRVEC
9317 033144 010137 033166 MOV R1,$LSTAD ;;LAST ADDRESS
9318 033150 010237 033170 MOV R2,$LSTBK ;;LAST BANK
9319 033154 012603 MOV (SP)+,R3 ;;RESTORE R3
9320 033156 012602 MOV (SP)+,R2 ;;RESTORE R2
9321 033160 012601 MOV (SP)+,R1 ;;RESTORE R1
9322 033162 012600 MOV (SP)+,RO ;;RESTORE RO
9323 033164 000207 RTS PC
9324 033166 000000 $LSTAD: .WORD 0 ;;CONTAINS THE LAST ADDRESS
9325 033170 000000 $LSTBK: .WORD 0 ;;CONTAINS THE LAST BANK
9326 .SBTTL SCOPE HANDLER ROUTINE
9327
9328 *****
9329 *THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
9330 *AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
9331 *AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
9332 *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9333 *SW14=1 LOOP ON TEST
9334 *SW11=1 INHIBIT ITERATIONS
9335 *SW09=1 LOOP ON ERROR
9336 *SW08=1 LOOP ON TEST IN SWR<7:0>
9337 *CALL
9338 * SCOPE ;;SCOPE=IOT
9339
9340 $SCOPE:
9341 033172 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
9342 ;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
9343 ;;OTHERWISE CONTINUE
9344 033174 021627 001002 CMP (SP),#1002 ;;UNEXPECTED TRAP OR INTERRUPT
9345 033200 101002 BHI 1$ ;;ARE TRAPPED HERE VIA IOT
9346 033202 000137 034212 JMP $ERROR ;;GO PROCESS UNEXPECTED TRAP
9347 033206 032777 040000 145724 1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
9348 033214 001131 BNE $OVER ;;YES IF SW14=1
9349 *****START OF CODE FOR THE XOR TESTER*****
9350 033216 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
9351 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
9352 033220 013746 000004 MOV $ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
9353 033224 012737 033244 000004 MOV #5,$ERRVEC ;;SET FOR TIMEOUT
9354 033232 005737 177060 TST $177060 ;;TIME OUT ON XOR?
9355 033236 012637 000004 MOV (SP)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
9356 033242 000500 BR $SVLAD ;;GO TO THE NEXT TEST
9357 033244 022626 $$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
9358 033246 012637 000004 MOV (SP)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
9359 033252 000440 BR 7$ ;;LOOP ON THE PRESENT TEST

```

```

9360 033254          6$:;#####END OF CODE FOR THE XOR TESTER#####
9361 033254 032777 000400 145656      BIT      #BIT08,@SWR      ;; LOOP ON SPEC. TEST?
9362 033262 001421          BEQ      2$          ;; BR IF NO
9363 033264 005046          CLR      -(SP)       ;; CLEAR A TEMP. LOCATION
9364 033266 117716 145646      MOVB    @SWR,(SP)    ;; PICKUP THE DESIRED TEST NUMBER
9365 033272 001414          BEQ      8$          ;; BRANCH IF BAD TEST NUMBER IN SWR
9366 033274 022716 000111      CMP      #11i,(SP)   ;; CHECK THE NUMBER IN THE SWR
9367 033300 002411          BLT      8$          ;; BRANCH IF TEST NUMBER IS OUT OF RANGE
9368 033302 011637 001102      MOV     (SP),$STNM   ;; UPDATE THE TEST NUMBER
9369 033306 005316          DEC     (SP)        ;; BACKUP BY ONE
9370 033310 006316          ASL     (SP)        ;; SCALE THE TEST NUMBER AS AN INDEX
9371 033312 062716 033516      ADD     $$SWO8TBL,(SP) ;; FORM THE ADDRESS OF TEST POINTER
9372 033316 013637 001106      MOV     @($SP)+,$LPADR ;; SET LOOP ADDRESS TO DESIRED TEST
9373 033322 000466          BR      $OVER       ;; GO LOOP ON THE TEST
9374 033324 005726          (SP)+    ;; CLEAN THE BAD TEST NUMBER OFF OF THE STACK
9375 033326 105737 001103      TSTB   $ERFLG      ;; HAS AN ERROR OCCURRED?
9376 033332 001421          BEQ      3$          ;; BR IF NO
9377 033334 123737 001115 001103      CMPB   $ERMAX,$ERFLG ;; MAX. ERRORS FOR THIS TEST OCCURRED?
9378 033342 101015          BHI     3$          ;; BR IF NO
9379 033344 032777 001000 145566      BIT      #BIT09,@SWR ;; LOOP ON ERROR?
9380 033352 001404          BEQ      4$          ;; BR IF NO
9381 033354 013737 001110 001106      7$:    MOV     $LPERR,$LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
9382 033362 000446          BR      $OVER
9383 033364 105037 001103      4$:    CLRB   $ERFLG      ;; ZERO THE ERROR FLAG
9384 033370 005037 001262          CLR     $TIMES      ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
9385 033374 000415          BR      1$          ;; ESCAPE TO THE NEXT TEST
9386 033376 032777 004000 145534      3$:    BIT      #BIT11,@SWR ;; INHIBIT ITERATIONS?
9387 033404 001011          BNE     1$          ;; BR IF YES
9388 033406 005737 001304          TST     $PASS       ;; IF FIRST PASS OF PROGRAM
9389 033412 001406          BEQ     1$          ;; INHIBIT ITERATIONS
9390 033414 005237 001104          INC     $ICNT       ;; INCREMENT ITERATION COUNT
9391 033420 023737 001262 001104      CMP     $TIMES,$ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
9392 033426 002024          BGE     $OVER       ;; BR IF MORE ITERATION REQUIRED
9393 033430 012737 000001 001104      1$:    MOV     #1,$ICNT   ;; REINITIALIZE THE ITERATION COUNTER
9394 033436 013737 033514 001262      MOV     $MXCNT,$TIMES ;; SET NUMBER OF ITERATIONS TO DO
9395 033444 105237 001102          $SVLAD: INCB   $STNM    ;; COUNT TEST NUMBERS
9396 033450 113737 001102 001302      MOVB   $STNM,$STNM  ;; SET TEST NUMBER IN APT MAILBOX
9397 033456 011637 001106          MOV     (SP),$LPADR ;; SAVE SCOPE LOOP ADDRESS
9398 033462 011637 001110          MOV     (SP),$LPERR ;; SAVE ERROR LOOP ADDRESS
9399 033466 005037 001264          CLR     $ESCAPE     ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
9400 033472 112737 000001 001115      MOVB   #1,$ERMAX    ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9401 033500 013777 001102 145434      $OVER: MOV     $STNM,@DISPLAY ;; DISPLAY TEST NUMBER
9402 033506 013716 001106          MOV     $LPADR,(SP) ;; FUDGE RETURN ADDRESS
9403 033512 000002          RTI                    ;; FIXES PS
9404 033514 003720          $MXCNT: 2000.        ;; MAX. NUMBER OF ITERATIONS
9405 033516          $$SWO8TBL:
9406 033516 003206          .WORD  TST1+2      ;; STARTING ADDRESS OF TEST 1
9407 033520 003324          .WORD  TST2+2      ;; STARTING ADDRESS OF TEST 2
9408 033522 003412          .WORD  TST3+2      ;; STARTING ADDRESS OF TEST 3
9409 033524 004642          .WORD  TST4+2      ;; STARTING ADDRESS OF TEST 4
9410 033526 004750          .WORD  TST5+2      ;; STARTING ADDRESS OF TEST 5
9411 033530 005126          .WORD  TST6+2      ;; STARTING ADDRESS OF TEST 6
9412 033532 005254          .WORD  TST7+2      ;; STARTING ADDRESS OF TEST 7
9413 033534 005360          .WORD  TST10+2     ;; STARTING ADDRESS OF TEST 10
9414 033536 006304          .WORD  TST11+2     ;; STARTING ADDRESS OF TEST 11
9415 033540 006422          .WORD  TST12+2     ;; STARTING ADDRESS OF TEST 12

```


9416	033542	006540	.WORD	TST13+2	:: STARTING ADDRESS OF TEST 13
9417	033544	007002	.WORD	TST14+2	:: STARTING ADDRESS OF TEST 14
9418	033546	007134	.WORD	TST15+2	:: STARTING ADDRESS OF TEST 15
9419	033550	007344	.WORD	TST16+2	:: STARTING ADDRESS OF TEST 16
9420	033552	007534	.WORD	TST17+2	:: STARTING ADDRESS OF TEST 17
9421	033554	007670	.WORD	TST20+2	:: STARTING ADDRESS OF TEST 20
9422	033556	010536	.WORD	TST21+2	:: STARTING ADDRESS OF TEST 21
9423	033560	011134	.WORD	TST22+2	:: STARTING ADDRESS OF TEST 22
9424	033562	011346	.WORD	TST23+2	:: STARTING ADDRESS OF TEST 23
9425	033564	011564	.WORD	TST24+2	:: STARTING ADDRESS OF TEST 24
9426	033566	011710	.WORD	TST25+2	:: STARTING ADDRESS OF TEST 25
9427	033570	012042	.WORD	TST26+2	:: STARTING ADDRESS OF TEST 26
9428	033572	012220	.WORD	TST27+2	:: STARTING ADDRESS OF TEST 27
9429	033574	012352	.WORD	TST30+2	:: STARTING ADDRESS OF TEST 30
9430	033576	012646	.WORD	TST31+2	:: STARTING ADDRESS OF TEST 31
9431	033600	013042	.WORD	TST32+2	:: STARTING ADDRESS OF TEST 32
9432	033602	013276	.WORD	TST33+2	:: STARTING ADDRESS OF TEST 33
9433	033604	013464	.WORD	TST34+2	:: STARTING ADDRESS OF TEST 34
9434	033606	013716	.WORD	TST35+2	:: STARTING ADDRESS OF TEST 35
9435	033610	014104	.WORD	TST36+2	:: STARTING ADDRESS OF TEST 36
9436	033612	014326	.WORD	TST37+2	:: STARTING ADDRESS OF TEST 37
9437	033614	014514	.WORD	TST40+2	:: STARTING ADDRESS OF TEST 40
9438	033616	014706	.WORD	TST41+2	:: STARTING ADDRESS OF TEST 41
9439	033620	015102	.WORD	TST42+2	:: STARTING ADDRESS OF TEST 42
9440	033622	015306	.WORD	TST43+2	:: STARTING ADDRESS OF TEST 43
9441	033624	015512	.WORD	TST44+2	:: STARTING ADDRESS OF TEST 44
9442	033626	016352	.WORD	TST45+2	:: STARTING ADDRESS OF TEST 45
9443	033630	016556	.WORD	TST46+2	:: STARTING ADDRESS OF TEST 46
9444	033632	016762	.WORD	TST47+2	:: STARTING ADDRESS OF TEST 47
9445	033634	017200	.WORD	TST50+2	:: STARTING ADDRESS OF TEST 50
9446	033636	017404	.WORD	TST51+2	:: STARTING ADDRESS OF TEST 51
9447	033640	017610	.WORD	TST52+2	:: STARTING ADDRESS OF TEST 52
9448	033642	017734	.WORD	TST53+2	:: STARTING ADDRESS OF TEST 53
9449	033644	020060	.WORD	TST54+2	:: STARTING ADDRESS OF TEST 54
9450	033646	020204	.WORD	TST55+2	:: STARTING ADDRESS OF TEST 55
9451	033650	020330	.WORD	TST56+2	:: STARTING ADDRESS OF TEST 56
9452	033652	020454	.WORD	TST57+2	:: STARTING ADDRESS OF TEST 57
9453	033654	020526	.WORD	TST60+2	:: STARTING ADDRESS OF TEST 60
9454	033656	020606	.WORD	TST61+2	:: STARTING ADDRESS OF TEST 61
9455	033660	021072	.WORD	TST62+2	:: STARTING ADDRESS OF TEST 62
9456	033662	021254	.WORD	TST63+2	:: STARTING ADDRESS OF TEST 63
9457	033664	021662	.WORD	TST64+2	:: STARTING ADDRESS OF TEST 64
9458	033666	022050	.WORD	TST65+2	:: STARTING ADDRESS OF TEST 65
9459	033670	022554	.WORD	TST66+2	:: STARTING ADDRESS OF TEST 66
9460	033672	022772	.WORD	TST67+2	:: STARTING ADDRESS OF TEST 67
9461	033674	023076	.WORD	TST70+2	:: STARTING ADDRESS OF TEST 70
9462	033676	023520	.WORD	TST71+2	:: STARTING ADDRESS OF TEST 71
9463	033700	024142	.WORD	TST72+2	:: STARTING ADDRESS OF TEST 72
9464	033702	024252	.WORD	TST73+2	:: STARTING ADDRESS OF TEST 73
9465	033704	025120	.WORD	TST74+2	:: STARTING ADDRESS OF TEST 74
9466	033706	025612	.WORD	TST75+2	:: STARTING ADDRESS OF TEST 75
9467	033710	026034	.WORD	TST76+2	:: STARTING ADDRESS OF TEST 76
9468	033712	026312	.WORD	TST77+2	:: STARTING ADDRESS OF TEST 77
9469	033714	026372	.WORD	TST100+2	:: STARTING ADDRESS OF TEST 100
9470	033716	026564	.WORD	TST101+2	:: STARTING ADDRESS OF TEST 101
9471	033720	027010	.WORD	TST102+2	:: STARTING ADDRESS OF TEST 102

9472	033722	027136		.WORD	TST103+2	:: STARTING ADDRESS OF TEST	103
9473	033724	027460		.WORD	TST104+2	:: STARTING ADDRESS OF TEST	104
9474	033726	030212		.WORD	TST105+2	:: STARTING ADDRESS OF TEST	105
9475	033730	030454		.WORD	TST106+2	:: STARTING ADDRESS OF TEST	106
9476	033732	030700		.WORD	TST107+2	:: STARTING ADDRESS OF TEST	107
9477	033734	031120		.WORD	TST110+2	:: STARTING ADDRESS OF TEST	110
9478	033736	031626		.WORD	TST111+2	:: STARTING ADDRESS OF TEST	111
9479	033740	032424		.WORD	\$EOP+2	:: ADDRESS OF END OF PASS	
9480	033742	044550		.WORD	ABTFAIL+2	:: ADDRESS OF ABORT FAILURE HANDLER	

.SBTTL APT COMMUNICATIONS ROUTINE

```

*****
9485 033744 112737 000001 034210 $ATY1: MOV  #1,$FFLG ;; TO REPORT FATAL ERROR
9486 033752 112737 000001 034206 $ATY3: MOV  #1,$MFLG ;; TO TYPE A MESSAGE
9487 033760 000403          $ATYC: BR          ;;
9488 033762 112737 000001 034210 $ATY4: MOV  #1,$FFLG ;; TO ONLY REPORT FATAL ERROR
9489 033770 $ATYC:
9490 033770 010046          MOV  RD,-(SP) ;; PUSH RD ON STACK
9491 033772 010146          MOV  R1,-(SP) ;; PUSH R1 ON STACK
9492 033774 105737 034206          TSTB $MFLG ;; SHOULD TYPE A MESSAGE?
9493 034000 001450          BEQ  5$ ;; IF NOT: BR
9494 034002 122737 000001 001316          CMPB #APTENV,$ENV ;; OPERATING UNDER APT?
9495 034010 001031          BNE  3$ ;; IF NOT: BR
9496 034012 132737 000100 001317          BITB #APTPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
9497 034020 001425          BEQ  3$ ;; IF NOT: BR
9498 034022 017600 000004          MOV  #4(SP),RD ;; GET MESSAGE ADDR.
9499 034026 062766 000002 000004          ADD  #2,4(SP) ;; BUMP RETURN ADDR.
9500 034034 005737 001276          1$: TST  $MSGTYPE ;; SEE IF DONE W/ LAST XMISSION?
9501 034040 001375          BNE  1$ ;; IF NOT: WAIT
9502 034042 010037 001312          MOV  RD,$MSGAD ;; PUT ADDR IN MAILBOX
9503 034046 105720          2$: TSTB (RD)+ ;; FIND END OF MESSAGE
9504 034050 001376          BNE  2$
9505 034052 163700 001312          SUB  $MSGAD,RD ;; SUB START OF MESSAGE
9506 034056 006200          ASR  RD ;; GET MESSAGE LNTH IN WORDS
9507 034060 010037 001314          MOV  RD,$MSGGLT ;; PUT LENGTH IN MAILBOX
9508 034064 012737 000004 001276          MOV  #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
9509 034072 000413          BR  5$
9510 034074 017637 000004 034120 3$: MOV  #4(SP),4$ ;; PUT MSG ADDR IN JSR LINKAGE
9511 034102 062766 000002 000004          ADD  #2,4(SP) ;; BUMP RETURN ADDRESS
9512 034110 013746 177776          MOV  177776,-(SP) ;; PUSH 177776 ON STACK
9513 034114 004737 044560          JSR  PC,$TYPE ;; CALL TYPE MACRO
9514 034120 000000          4$: .WORD  D
9515 034122          5$:
9516 034122 105737 034210          10$: TSTB $FFLG ;; SHOULD REPORT FATAL ERROR?
9517 034126 001416          BEQ  12$ ;; IF NOT: BR
9518 034130 005737 001316          TST  $ENV ;; RUNNING UNDER APT?
9519 034134 001413          BEQ  12$ ;; IF NOT: BR
9520 034136 005737 001276          11$: TST  $MSGTYPE ;; FINISHED LAST MESSAGE?
9521 034142 001375          BNE  11$ ;; IF NOT: WAIT
9522 034144 017637 000004 001300          MOV  #4(SP),$FATAL ;; GET ERROR #
9523 034152 062766 000002 000004          ADD  #2,4(SP) ;; BUMP RETURN ADDR.
9524 034160 005237 001276          INC  $MSGTYPE ;; TELL APT TO TAKE ERROR
9525 034164 105037 034210          12$: CLRB $FFLG ;; CLEAR FATAL FLAG
9526 034170 105037 034207          CLRB $LFLG ;; CLEAR LOG FLAG
9527 034174 105037 034206          CLRB $MFLG ;; CLEAR MESSAGE FLAG

```

```

9528 034200 012601          MOV      (SP)+,R1          ;; POP STACK INTO R1
9529 034202 012600          MOV      (SP)+,R0          ;; POP STACK INTO R0
9530 034204 000207          RTS      PC              ;; RETURN
9531 034206 000          $MFLG: .BYTE 0          ;; MESSG. FLAG
9532 034207 000          $LFLG: .BYTE 0          ;; LOG FLAG
9533 034210 000          $FFLG: .BYTE 0          ;; FATAL FLAG
9534          034212          .EVEN
9535          000200          APTSIZE=200
9536          000001          APTENV=001
9537          000100          APTSPool=100
9538          000040          APTCSUP=040
9539          .SBTTL  ERROR HANDLER ROUTINE
9540
9541          ;; *****
9542          ;; *THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9543          ;; *SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9544          ;; *AND GO TO TYPERR ON ERROR
9545          ;; *THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9546          ;; *SW15=1          HALT ON ERROR
9547          ;; *SW13=1          INHIBIT ERROR TYPEOUTS
9548          ;; *SW10=1         BELL ON ERROR
9549          ;; *SW09=1         LOOP ON ERROR
9550          ;; *CALL
9551          ;; *      ERROR      N          ;;;ERROR=EMT AND N=ERROR ITEM NUMBER
9552
9553          $ERROR:
9554          034212 104407          CKSWR          ;; TEST FOR CHANGE IN SOFT-SWR
9555          034214 105237 001103          7$: INCB      $ERFLG          ;; SET THE ERROR FLAG
9556          034220 001775          BEQ      7$          ;; DON'T LET THE FLAG GO TO ZERO
9557          034222 013777 001102 144712          MOV      $STNM,$DISPLAY ;; DISPLAY TEST NUMBER AND ERROR FLAG
9558          034230 032777 002000 144702          BIT      #BIT10,$SWR      ;; BELL ON ERROR?
9559          034236 001402          BEQ      1$          ;; NO - SKIP
9560          034240 104401 001266          TYPE     $BELL          ;; RING BELL
9561          034244 005237 001112          1$: INC      $ERTTL        ;; COUNT THE NUMBER OF ERRORS
9562          034250 011637 001116          MOV      (SP) $ERRPC      ;; GET ADDRESS OF ERROR INSTRUCTION
9563          034254 162737 000002 001116          SUB      #2,$ERRPC
9564          034262 117737 144630 001114          MOVB    @ $ERRPC,$ITEMB  ;; STRIP AND SAVE THE ERROR ITEM CODE
9565          034270 032777 020000 144642          BIT      #BIT13,$SWR      ;; SKIP TYPEOUT IF SET
9566          034276 001055          BNE     20$          ;; SKIP TYPEOUTS
9567          034300 021627 001002          CMP      (SP),#1002      ;; IF RETURN PC LESS THAN 1002
9568          034304 101046          BHI     12$          ;; ERROR IS ILLEGAL TRAP
9569          ;;;PROCESS UNEXPECTED TRAP OR INTERRUPT
9570          034306 016637 000004 001116          MOV      4(SP) $ERRPC    ;; GET PC AT TIME OF FALSE TRAP
9571          034314 162737 000002 001116          SUB      #2,$ERRPC      ;; ADJUST PC
9572          034322 104401 034366          TYPE     10$          ;; TYPE HEADER
9573          034326 013746 001116          MOV      $ERRPC,-(SP)   ;; SAVE $ERRPC FOR TYPEOUT
9574          034332 104402          TYP0C          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9575          034334 104401 034374          TYPE     11$
9576          034340 162716 000004          SUB      #4,(SP)        ;; GET FALSE TRAP VECTOR ADDR
9577          034344 011637 001116          MOV      (SP) $ERRPC
9578          034350 013746 001116          MOV      $ERRPC,-(SP)  ;; SAVE $ERRPC FOR TYPEOUT
9579          034354 104402          TYP0C          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
9580          034356 104401 001273          TYPE     $CRLF
9581          034362 022626          CMP      (SP)+,(SP)+   ;; POP FALSE TRAP VECTOR PC&ADDR
9582          034364 000422          BR      20$
9583          034366 050200 036503 000040 10$: .ASCIZ <200>'PC= '

```

```

9584 034374 020040 047125 054105 11$: .ASCIZ ' UNEXPECTED TRAP TO '
9585 034402 042520 052103 042105
9586 034410 052040 040522 020120
9587 034416 047524 000040
9588
9589 034422 .EVEN
9590 034422 12$: JSR PC,TYPERR ;;GO TO USER ERROR ROUTINE
9591 034426 004737 034534 TYPE ,SCRLF
9592 034432 104401 001273
9593 034432 122737 000001 001316 20$: CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
9594 034440 001007 BNE 2$ ;;NO SKIP APT ERROR REPORT
9595 034442 113737 001114 034454 MOVB $ITEMB,21$ ;;SET ITEM NUMBER AS ERROR NUMBER
9596 034450 004737 033762 JSR PC,$ATY4 ;;REPORT FATAL ERROR TO APT
9597 034454 000 21$: .BYTE 0
9598 034455 000 .BYTE 0
9599 034456 000777 22$: BR 22$ ;;APT ERROR LOOP
9600 034460 005777 144454 2$: TST @SWR ;;HALT ON ERROR
9601 034464 100002 BPL 3$ ;;SKIP IF CONTINUE
9602 034466 000000 HALT ;;HALT ON ERROR!
9603 034470 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
9604 034472 032777 001000 144440 3$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
9605 034500 001402 BEQ 4$ ;;BR IF NO
9606 034502 013716 001110 MOV $LPERR,(SP) ;;FUDGE RETURN FOR LOOPING
9607 034506 005737 001264 4$: TST $ESCAPEE ;;CHECK FOR AN ESCAPE ADDRESS
9608 034512 001402 BEQ 5$ ;;BR IF NONE
9609 034514 013716 001264 MOV $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
9610 034520 5$:
9611 034520 022737 032612 000042 CMP #SENDAD,@#42 ;;ACT-11 AUTO-ACCEPT?
9612 034526 001001 BNE 6$ ;;BRANCH IF NO
9613 034530 000000 HALT ;;YES
9614 034532 6$:
9615 034532 000002 RTI ;;RETURN
9616 *****
9617 .SBTTL TYPE ERROR ROUTINE
9618 *ENTRY JSR PC,TYPERR
9619 *RETURN RTS PC
9620 *
9621 *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9622 *ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
9623 *ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
9624 *THE ERROR.
9625 *****
9626 †TYPERR: SAVREG
9627 034534 104413 MOVB $STNM,$STN ;GET TEST NUMBER FOR REPORT
9628 034536 113737 001102 001302 BIC #177400,$STN ;CLEAR UNUSED BITS
9629 034544 042737 177400 001302 MOVB $ITEMB,R0 ;ENTER ERROR NUMBER
9630 034552 113700 001114 MOVB #177400,R0 ;CLEAR UNUSED BITS
9631 034556 042700 177400 BIC #177400,R0 ;FORM INDEX FOR ERROR TABLE
9632 034562 005300 DEC R0
9633 034564 006300 ASL R0
9634 034566 006300 ASL R0
9635 034570 006300 ASL R0
9636 034572 062700 001360 1$: ADD #ERRTB,R0 ;FORM ADDRESS OF ERROR ENTRY
9637 034576 012037 034612 MOV (R0)+,2$ ;GET EM POINTER
9638 034602 001404 BEQ 3$ ;BRANCH IF THERE ISN'T ONE
9639 034604 104401 001273 TYPE ,SCRLF ;TYPE CARRIAGE RETURN LINE FEED
9640 034610 104401 TYPE ;TYPE ERROR MESSAGE (EM)

```

```

9640 034612 000000          2$: .WORD 0          ;EM POINTER GOES HERE
9641 034614 012037 034630  3$: MOV (R0)+,4$      ;GET DH POINTER
9642 034620 001404          BEQ 5$              ;BRANCH IF THERE ISN'T ONE
9643 034622 104401 001273  TYPE , $CRLF      ;TYPE CR-LF
9644 034626 104401          TYPE                ;TYPE DATA HEADER
9645 034630 000000          4$: .WORD 0          ;DH POINTER GOES HERE
9646 034632 012001          5$: MOV (R0)+,R1      ;GET DT POINTER
9647 034634 001445          BEQ 20$            ;BRANCH IF THERE ARE NONE
9648 034636 005004          CLR R4             ;SET INDENT SWITCH
9649 034640 012000          MOV (R0)+,R0      ;GET DF POINTER
9650 034642 012002          MOV (R0)+,R2      ;STORE NUMBER OF DH'S
9651 034644 104401 001273  TYPE , $CRLF
9652 034650 112003          10$: MOV#B (R0)+,R3    ;GET & STORE NUMBER OF DATA WORDS
9653 034652 105720          TSTB (R0)+        ;BUMP PAST FORMAT WORD
9654 034654 005703          TST R3            ;TEST IF ANY DATA FOR THIS HEADER
9655 034656 001416          BEQ 14$           ;NO - SKIP DATA PRINT
9656 034660 005704          TST R4           ;CHECK FOR INDENT
9657 034662 001004          BNE 12$          ;YES, GO INDENT
9658 034664 013146          11$: MOV @ (R1)+, -(SP) ;PUT FIRST DATA WORD ON STACK
9659 034666 104402          TYPOC           ;TYPE IT
9660 034670 005303          DEC R3           ;MORE DATA WORDS
9661 034672 001403          BEQ 13$          ;NO-BRANCH
9662 034674 104401 051341  12$: TYPE ,SPACE2   ;TYPE SEPARATORS
9663 034700 000771          BR 11$           ;LOOP
9664 034702 104401 001273  13$: TYPE , $CRLF    ;TYPE <CR><LF>
9665 034706 005710          TST (R0)         ;CHECK IF NEXT HEADER AVAILABLE
9666 034710 001401          BEQ 14$          ;NO, DO NOT CHANGE INDENT
9667 034712 005104          COM R4          ;CHANGE INDENT
9668 034714 005302          14$: DEC R2       ;MORE DH'S?
9669 034716 003414          BLE 20$         ;NO-BRANCH
9670 034720 012037 034740  15$: MOV (R0)+,18$   ;GET NEXT DH POINTER
9671 034724 001751          BEQ 10$          ;IF 0 GET DATA
9672 034726 005704          TST R4          ;INDENT?
9673 034730 001402          BEQ 17$         ;NO, BRANCH
9674 034732 104401 051341  TYPE ,SPACE2     ;TYPE INDENT
9675 034736 104401          17$: TYPE ,DH    ;TYPE DH
9676 034740 000000          18$: .WORD 0      ;DH POINTER GOES HERE
9677 034742 104401 001273  TYPE , $CRLF
9678 034746 000740          BR 10$          ;GET DATA
9679 034750 104414          20$: RESREG
9680 034752 005237 001632  INC ERRCNT       ;INCREMENT THE ERROR COUNT
9681 034756 032777 010000 144154  BIT #SW12, @SWR  ;CHECK IF SWITCH 12 SET
9682 034764 001421          BEQ 25$         ;NO, RETURN
9683 034766 022737 000024 001632  CMP #20.,ERRCNT ;CHECK IF ERROR THRESHOLD EXCEEDED
9684 034774 103015          BHS 25$         ;NO, RETURN
9685 034776 104401 053661  TYPE ,ABORT     ;TYPE "PROGRAM ABORTED BECAUSE
9686                                     ;ERROR THRESHOLD EXCEEDED"
9687 035002 005737 000042  TST 42           ;CHECK IF CHAIN MODE
9688 035006 001407          BEQ 22$         ;NO, HALT PROCESSOR
9689 035010 012706 001100  MOV #STACK, SP  ;INITIALIZE STACK
9690 035014 012737 000001 032450  MOV #1, $EOPCT  ;FORCE END OF PROGRAM
9691 035022 000137 032422  JMP $EOP
9692 035026 000000          22$: HALT
9693 035030 000207          25$: RTS PC
9694                                     ;.SBTTL
9695                                     ;*

```

```

NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER
THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.

```

```

9696      ;*      THIS IS THE INDICATION TO THE ROUTINE CHECKING
9697      ;*      NON-EXISTANT MEMORY THAT AN INTERRUPT DID OCCUR.
9698
9699      035032 005237 001662      NEXINT: INC      INTSET      ;BUMP THE INTERRUPT COUNTER
9700      035036 000002
9701
9702      .SBTTL  RK611 INTERRUPT HANDLER
9703      ;*      MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE.  ACTUAL
9704      ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
9705      ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
9706      ;*      INTERRUPT OCCURRED.
9707
9708      035040 000240      INTHLR: NOP
9709      035042 005237 001662      INC      INTSET      ;BUMP THE INTERRUPT FLAG
9710      035046 000002      RTI      ;RETURN.
9711
9712      .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
9713      ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE.  THE REPORT
9714      ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
9715
9716      035050 032777 020000 144062  PERHLR: BIT      #BIT13, @SWR      ;TEST IF INHIBIT REPORT
9717      035056 001003      BNE      1$      ;YES - SKIP
9718      035060 104401 054110      TYPE      ,EM3      ;TYPE PARITY ERROR MESSAGE
9719      035064 104402      TYPOC      ;AND PC VALUE
9720      035066 012737 000001 032450  1$:  MOV      #1, $EOPCT      ;FORCE END OF PROGRAM
9721      035074 012706 001100      MOV      #STACK, SP      ;CLEAN OFF STACK
9722      035100 000137 032422      JMP      $EOP
9723
9724      .SBTTL  LINE CLOCK INTERRUPT HANDLER
9725      ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
9726      ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
9727
9728      035104 005237 001674      LCKHLR: INC      LCLKTK      ;INCREMENT CLOCK TICK COUNTER
9729      035110 042777 000200 144572  BIC      #BIT7, @KWLADD      ;CLEAR MONITOR BIT
9730      035116 000002      RTI
9731
9732      ;*****
9733      .SBTTL  OPTION PRESENT TEST AND SETUP
9734      ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
9735      ;*      LINE CLOCK ARE ON THE SYSTEM.  FLAGS ARE SET IF PRESENT;  CLEARED
9736      ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
9737      035120 104413      OPTTST: SAVREG
9738      035122 013746 000004      MOV      ERRVEC, -(SP)      ;STORE OLD NEM CONTENTS
9739      035126 013746 000006      MOV      ERRVEC+2, -(SP)
9740      035132 012737 035330 000004      MOV      #20$, ERRVEC      ;DET INTERRUPT FOR NEM
9741      035140 012737 000340 000006      MOV      #PR7, ERRVEC+2      ;SET PRIORITY
9742      035146 005037 001670      CLR      MEMPAR      ;CLEAR PARITY WORD FLAGS
9743      035152 042737 000100 001664      BIC      #PARPRE, OPTFLG      ;CLEAR PARITY PRESENT FLAG
9744      035160 005737 170200      TST      170200      ;TEST IF 11/70 REGISTER PRESENT
9745      035164 000240      NOP
9746      035166 000240      NOP
9747      035170 012737 177750 001672      MOV      #177750, CSRPTR      ;SET POINTER FOR 11/70
9748      035176 052737 002000 001664      BIS      #CP1170, OPTFLG      ;SET 11/70 FLAG
9749      035204 052737 000100 001664      BIS      #PARPRE, OPTFLG      ;SET PARITY PRESENT FLAG
9750      035212 000503      BR      35$      ;GO TO VECTOR SETUP
9751

```

9752	035214	013703	001714	3\$:	MOV	MEMBAS,R3	;	SET UP POINTER TO FIRST PARITY CSR	
9753	035220	012704	000001		MOV	#1,R4	;	INIT MASK	
9754	035224	012713	000001	6\$:	MOV	#1,(R3)	;	SET PARITY DETECT IN THAT CSR	
9755	035230	005713			TST	(R3)			
9756	035232	050437	001670		BIS	R4,MEMPAR	;	SET PARITY PRESENT BIT	
9757				:	BIT	#PARPRE,OPTFLG	;	WAS PARITY PRESENT SET BEFORE	
9758				:	BNE	10\$;	YES - SKIP	
9759	035236	013700	001716		MOV	MMVECA,RO	;	SET UP FOR PARITY TRAPS	
9760	035242	012720	035342		MOV	#40\$,(RO)+	;	TO 40\$	
9761	035246	012710	000340		MOV	#PR7,(RO)			
9762	035252	012700	072640		MOV	#OBUFF+224,RO	;	SET POINTER TO BUFFER WHERE BAD PARITY	
9763							;	IS USED IN THE TESTS	
9764	035256	012713	000004		MOV	#BIT2,(R3)	;	SET TO WRITE WRONG PARITY	
9765	035262	012710	177777		MOV	#-1,(RO)	;	WRITE WORD BAD	
9766	035266	012713	000001		MOV	#1,(R3)	;	SET TO DETECT PARITY ERROR	
9767	035272	005710			TST	(RO)	;	READ BAD WORD	
9768	035274	042737	000100	001664	BIC	#PARPRE,OPTFLG	;	REV 005	
9769	035302	012713	000002		MOV	#2,(R3)	;	TRUN OFF CZR ERROR ENABLE	
9770	035306	012710	000000		MOV	#0,(RO)	;	CLEAR BAD PARITY	
9771				:	BIC	#7,(R3)	;	REV 006	
9772				:	MOV	#0,(RO)	;	REV 006	
9773	035312	000443			BR	35\$;	REV 005	
9774	035314	062703	000002	10\$:	ADD	#2,R3	;	BUMP TO NEXT CSR	
9775	035320	000241			CLC				
9776	035322	006104			ROL	R4	;	SHIFT MASK	
9777	035324	001337			BNE	6\$;	TEST IF ALL DONE	
9778	035326	000435			BR	35\$;	YES - SKIP	
9779									
9780	035330	022626		20\$:	CMP	(SP)+,(SP)+	;	CLEAR STACK	
9781	035332	012737	035416	000004	MOV	#30\$,ERRVEC	;	SET NEW NEM TRAP ADDRESS	
9782	035340	000725			BR	3\$;	GO TO CSR CHECKS	
9783									
9784	035342	022626		40\$:	CMP	(SP)+,(SP)+	;	CLEAR STACK	
9785	035344	010337	001672		MOV	R3,CSRPTR	;	SET CSR POINTER FOR CSR TO BE USED	
9786	035350	052737	000100	001664	BIS	#PARPRE,OPTFLG	;	SET PARITY PRESENT FLAG	
9787	035356	012713	000002		MOV	#2,(R3)	;	TRUN OFF CSR ERROR ENABLE	
9788	035362	012710	000000		MOV	#0,(RO)	;	WRITE GOOD PARITY	
9789	035366	032713	000002		BIT	#2,(R3)	;	IS THIS THE MFS11-K MEMORY ?	
9790	035372	001003			BNE	47\$;	BRANCH IF 50	
9791				:	CLR	(R3)	;	CLEAR CSR	
9792				:	CLR	(RO)	;	CLEAR BAD WORD	
9793	035374	012713	000001		MOV	#1,(R3)	;	SET TO DETECT PARITY ERRORS	
9794	035400	000745			BR	10\$;	CHECK NEXT CSR	
9795									
9796	035402	042737	000100	001664	47\$:	BIC	#PARPRE,OPTFLG	;	FORGET THEM ALL
9797	035410	005037	001670		CLR	MEMPAR	;	FORGET THEM ALL	
9798	035414	000402			BR	35\$;	EXIT	
9799	035416	022626		30\$:	CMP	(SP)+,(SP)+	;	CLEAR STACK	
9800	035420	000735			BR	10\$;	GO CHECK NEXT CSR	
9801	035422	013700	001716		35\$:	MOV	MMVECA,RO	;	SET UP POINTER TO PARITY VECTOR
9802	035426	005737	001670		TST	MEMPAR	;	TEST IF ANY PARITY PRESENT	
9803	035432	001004			BNE	37\$;	YES - SKIP	
9804	035434	032737	002000	001664	BIT	#CP1170,OPTFLG	;	TEST IF 11/70	
9805	035442	001405			BEQ	39\$;	NO - SKIP	
9806	035444	012720	035050		37\$:	MOV	#PERHLR,(RO)+	;	SET UP PARITY VECTOR
9807	035450	012710	000340		MOV	#PR7,(RO)	;	SET PRIORITY	

```

9808 035454 000403
9809 035456 012720 000116
9810 035462 005010
9811 035464 012737 035032 000004
9812 035472 005037 001662
9813 035476 013700 001712
9814 035502 012720 035104
9815 035506 012710 000340
9816 035512 012777 000100 144170
9817 035520 000240
9818 035522 000240
9819 035524 000240
9820 035526 005737 001662
9821 035532 001003
9822
9823 035534 052737 100000 001664
9824 035542 012701 000006
9825 035546 005037 001662
9826 035552 012637 000006
9827 035556 012637 000004
9828 035562 012746 000000
9829 035566 012746 035574
9830 035572 000002
9831 035574
9832 035574 104414
9833 035576 000207
9834
9835
9836
9837
9838
9839
9840
9841 035600 032777 001000 143332
9842 035606 001405
9843 035610 105737 001103
9844 035614 001402
9845 035616 013716 001110
9846 035622 000002
9847
9848
9849
9850
9851
9852
9853 035624 005037 001674
9854 035630 032737 100000 001664
9855 035636 001004
9856 035640 012737 000020 001654
9857 035646 000410
9858 035650 005737 001662
9859 035654 001005
9860 035656 005737 001674
9861 035662 001772
9862 035664 005037 001674
9863 035670 000207

39$: BR 38$
MOV #116,(R0)+ ;ELSE RESTORE TRAP CATCHER
CLR (R0)
38$: MOV #NEXINT,ERRVEC ;SET UP NEM VECTOR FOR LINE CLOCK TEST
CLR INTSET ;CLEAR INTERRUPT COUNTER
MOV KWLVEC,R0 ;SET POINTER TO VECTOR
MOV #LCKHLR,(R0)+ ;INSERT ADDRESS OF INTERRUPT HDLR
MOV #PR7,(R0) ;INSERT PRIORITY
MOV #BIT6,@KWLADD ;LOAD KW11-L FOR INTERRUPT ENABLE
NOP
NOP
NOP
TST INTSET ;TEST IN NEM ON KW11-P REFERENCE
BNE 4$ ;THIS BRANCH WILL BYPASS SET UP OF
;CLOCK OPTION
4$: BIS #LCLKPR,OPTFLG ;SET CLOCK PRESENT FLAG
MOV #6,R1 ;RESTORE OLD VECTOR
CLR INTSET ;CLEAR INT FLAG
MOV (SP)+,ERRVEC+2
MOV (SP)+,ERRVEC
MOV #PRO,-(SP) ;RESTORE PRIORITY TO 0
MOV #12$,-(SP)
RTI

12$: RESREG
RTS PC

;*****
;SBTTL LOOP ON INTERNAL ERROR
;* THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
;* IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
;* TO BE EXECUTED ON ERROR.
SCOP1$: BIT #SW9,@SWR ;CHECK IF LOOP ON ERROR
BEQ 5$ ;NO, CONTINUE
TSTB $ERFLG ;CHECK IF ERROR OCCURRED
BEQ 5$
MOV $LPERR,(SP) ;LOAD ERROR RETURN
5$: RTI ;RETURN

;SBTTL LINE CLOCK CALIBRATE
;* WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS
;* TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
;* THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
;* LINE CLOCK SIMULATOR.
CLKCAL: CLR LCLKTK ;CLEAR TICK COUNTER
BIT #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
BNE 1$ ;YES - SKIP
MOV #16.,TIMCNT ;ELSE PRESET TIMCNT
BR 2$ ;AND EXIT
1$: TST INTSET ;TEST IF INTERRUPT HAS OCCURRED
BNE 2$ ;YES- ABORT CALIBRATION
TST LCLKTK ;WAIT FOR CLOCK TICK
BEQ 1$ ;NOT YET - LOOP
CLR LCLKTK ;CLEAR TICK COUNT
2$: RTS PC ;RETURN

```



```

9920 036016 104413          IWAT128: SAVREG          ;ENTRY FOR 128 MS WAIT
9921 036020 012700 000010      MOV          #8,RO
9922 036024 000433          BR          WATSRT
9923 036026 104413          IWAT112: SAVREG          ;ENTRY FOR 112 MS WAIT
9924 036030 012700 000007      MOV          #7,RO
9925 036034 000427          BR          WATSRT
9926 036036 104413          IWAT96:  SAVREG          ;ENTRY FOR 96 MS WAIT
9927 036040 012700 000006      MOV          #6,RO
9928 036044 000423          BR          WATSRT
9929 036046 104413          IWAT80:  SAVREG          ;ENTRY FOR 80 MS WAIT
9930 036050 012700 000005      MOV          #5,RO
9931 036054 000417          BR          WATSRT
9932 036056 104413          IWAT64:  SAVREG          ;ENTRY FOR 64 MS WAIT
9933 036060 012700 000004      MOV          #4,RO
9934 036064 000413          BR          WATSRT
9935 036066 104413          IWAT48:  SAVREG          ;ENTRY FOR 48 MS WAIT
9936 036070 012700 000003      MOV          #3,RO
9937 036074 000407          BR          WATSRT
9938 036076 104413          IWAT32:  SAVREG          ;ENTRY FOR 32 MS WAIT
9939 036100 012700 000002      MOV          #2,RO
9940 036104 000403          BR          WATSRT
9941 036106 104413          IWAT16:  SAVREG          ;ENTRY FOR 16 MS WAIT
9942 036110 012700 000001      MOV          #1,RO
9943 036114 012746 000000      WATSRT:  MOV          #PRD,-(SP) ;LOAD PRIORITY 0 ON STACK
9944 036120 012746 036126      MOV          #5$,-(SP) ;LOAD ADDRESS
9945 036124 000002          RTI
9946
9947 036126 012737 000020 001654 5$:  MOV          #16,,TIMCNT ;PRESET TIME COUNTER
9948 036134 004737 035624          JSR          PC,CLKCAL ;GO CALIBRATE THE CLOCK
9949 036140 005737 001662          1$:  TST          INTSET ;TEST IF INTERRUPT OCCURRED
9950 036144 001036          BNE          3$ ;YES - EXIT
9951 036146 032737 100000 001664 BIT          #LCLKPR,OPTFLG ;TEST IF KW11-L AVAILABLE
9952 036154 001002          BNE          2$ ;YES - SKIP
9953 036156 004737 035672          JSR          PC,MYTIME ;ELSE CALL SIMULATOR
9954 036162 023700 001674          2$:  CMP          LCLKTK,RO ;TEST IF ENOUGH TICKS COUNTED
9955 036166 103764          BLO          1$ ;NO - LOOP
9956 036170 104420          TGETRK          ;ELSE GET '611 REGS
9957 036172 013701 001540          MOV          T,CS1,R1 ;PUT CS1 IN R1- STRIP ALL BUT
9958 036176 042701 177741          BIC          #177741,R1 ;COMMAND CODE; INDEX INTO TABLE
9959 036202 016137 050476 001372 MOV          CMNDLB(R1),DH2N ;AND SELECT HEADER TO IDENTIFY OPERATION
9960 036210 012737 054160 001370 MOV          #EM4,EM2N ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
9961 036216 013700 001302          MOV          $TESTN,RO ;GET NUMBER OF PRESENT TEST
9962 036222 006300          ASL          RO ;SHIFT FOR INDEX
9963 036224 016037 033516 001264 MOV          $$SWO8TB(RO),$ESCAPE ;LOAD ESCAPE TO ABORT TESTS
9964 036232 162737 000002 001264 SUB          #2,$ESCAPE ;BUT GO TO NEXT SCOPE CALL
9965 036240 000402          BR          4$
9966 036242 062716 000002          3$:  ADD          #2,(SP) ;BUMP RETURN AROUND ERROR
9967 036246 104414          4$:  RESREG          ;RESTORE REGS
9968 036250 000002          RTI          ;RETURN
9969
9970          .SBTTL          "L" REGISTER LOADING ROUTINE
9971          ;*          THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
9972          ;*          THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT
9973          ;*          LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
9974          ;*          FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
9975          ;*

```

```

9976          ;*      CALL:   JSR      R4,LRLoad
9977          ;*      COMMAND
9978          ;*      WORD COUNT
9979          ;*      BUS ADDRESS
9980          ;*      .BYTE          SECTOR ADDRESS
9981          ;*      .BYTE          TRACK ADDRESS
9982          ;*      CYLINDER ADDRESS
9983
9984 036252          LRLoad:  MOV      RO,-(SP)          ;; PUSH RO ON STACK
9985 036252 010046    MOV      R1,-(SP)          ;; PUSH R1 ON STACK
9986 036254 010146    MOV      #L.CS1,R1        ;; GET ADDRESS OF L REGS
9987 036256 012701 001600  MOV      #4,RO           ;; PRESET COUNT
9988 036262 012700 000004  1$:  MOV      (R4)+,(R1)+      ;; MOVE FIRST FOUR WORDS INTO "L" REGS
9989 036266 012421    DEC      RO              ;; CS1, WC, BA, DA
9990 036270 005300    BNE     1$
9991 036272 001375    MOV      DRVNUM,(R1)+     ;; LOAD DRIVE NUMBER
9992 036274 013721 001626  TST      (R1)+           ;; BUMP PAST ASOF
9993 036300 005721    MOV      (R4)+,(R1)      ;; LOAD DCYL
9994 036302 012411    MOV      (SP)+,R1        ;; POP STACK INTO R1
9995 036304 012601    MOV      (SP)+,RO        ;; POP STACK INTO RO
9996 036306 012600    RTS      R4
9997 036310 000204
9998
9999          .SBTTL  LOAD RK611 FOR OPERATION
10000         ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
10001         ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
10002         ;*      OF THE REGISTER CONTENTS. L.CS1 IS TRANSFERRED LAST AS IT
10003         ;*      SHOULD BE IF THE GO BIT IS SET.
10004 036312 005037 001662  LOADRK: CLR      INTSET          ;; CLEAR INTERRUPT FLAG.
10005 036316 010046    MOV      RO,-(SP)          ;; STORE REGISTER
10006 036320 010146    MOV      R1,-(SP)
10007 036322 012700 001602  MOV      #L.WC,RO         ;; GET ADDRESS OF SETUP STORAGE WC
10008 036326 010201    MOV      R2,R1           ;; GET BASE ADDRESS
10009 036330 062701 000002  ADD      #2,R1           ;; PUT R1 PAST RKCS1
10010 036334 012021    MOV      (RO)+,(R1)+      ;; LOAD WORD COUNT
10011 036336 012021    MOV      (RO)+,(R1)+      ;; LOAD BUS ADDRESS
10012 036340 012021    MOV      (RO)+,(R1)+      ;; LOAD DISK ADDRESS
10013 036342 012011    MOV      (RO)+,(R1)      ;; LOAD CS2
10014 036344 062701 000006  ADD      #6,R1           ;; BUMP R1 TO ASOF
10015 036350 012021    MOV      (RO)+,(R1)+      ;; LOAD OFFSET
10016 036352 012021    MOV      (RO)+,(R1)+      ;; LOAD CYLINDER
10017 036354 062701 000004  ADD      #4,R1           ;; BUMP R1 TO MR1
10018 036360 011011    MOV      (RO),(R1)        ;; LOAD MR1
10019 036362 123727 001102 000007  CMPB    $STNM,#7         ;; SEE IF TEST 7
10020 036370 001403    BEQ     1$              ;; BR IF YES
10021 036372 053737 001720 001600  1$:  BIS      DTYPE,L.CS1     ;; ELSE SET CORRECT DRIVE TYPE
10022 036400 013712 001600    MOV      L.CS1,(R2)      ;; LOAD CS1
10023 036404 012601    MOV      (SP)+,R1        ;; RESTORE REGISTER
10024 036406 012600    MOV      (SP)+,RO
10025 036410 000002    RTI
10026
10027
10028         ; THIS ROUTINE LOADS CONSTANTS EITHER FOR THR RK06 OR THE RK07
10029
10030 036412 005737 001720  LDCON: TST      DTYPE          ;; SEE IF RK06
10031 036416 001414    BEQ     1$              ;; BR IF YES

```

```

10032 036420 012737 001456 036476      MOV      #1456,LSTCYL      ;ELSE LOAD RK07 CONSTANTS
10033 036426 012737 001000 036500      MOV      #1000,TSTCYL    ;MAY NOT BE NEEDED
10034 036434 005037 036506      CLR      HOLD2
10035 036440 012737 000040 036510      MOV      #DTYE,HOLD3
10036 036446 000207      RTS      PC
10037
10038 036450 012737 000632 036476 1$:      MOV      #632,LSTCYL    ;RK06 CONSTANTS
10039 036456 005037 036500      CLR      TSTCYL        ;MAY NOT BE NEEDED
10040 036462 012737 002000 036506      MOV      #CDT,HOLD2
10041 036470 005037 036510      CLR      HOLD3
10042 036474 000207      RTS      PC
10043
10044 036476 000000      LSTCYL: 0
10045 036500 000000      TSTCYL: 0                ;FOR T20,57,60 MAY NOT BE NEEDED
10046 036502 000000      HOLD: 0                  ;FOR T73,105
10047 036504 000000      HOLD1: 0                 ;FOR T105
10048 036506 000000      HOLD2: 0                 ;FOR T7
10049 036510 000000      HOLD3: 0                 ;FOR T12
10050
10051
10052 ; THESE ROUTINES LOAD THE CORRECT DRIVE TYPE INTO DTYPE
10053 ;
10054 036512 006300      GTYP0:  ASL      R0
10055 036514 016037 001722 001720      MOV      TYPTBL(R0),DTYPE
10056 036522 006200      ASR      R0
10057 036524 000207      RTS      PC
10058
10059 036526 006301      GTYP1:  ASL      R1
10060 036530 016137 001722 001720      MOV      TYPTBL(R1),DTYPE
10061 036536 006201      ASR      R1
10062 036540 000207      RTS      PC
10063
10064
10065 .SBTTL  STORE RK611 REGISTERS
10066 ;*      ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
10067 ;*      WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN
10068 ;*      THIS ROUTINE.
10069
10070 036542 010046      GETRK:  MOV      R0,-(SP)      ;STORE REGISTERS TO BE USED
10071 036544 010146      MOV      R1,-(SP)
10072 036546 010346      MOV      R3,-(SP)
10073 036550 012700 001540      MOV      #T.CS1,R0      ;SET POINTER TO TEST TABLE
10074 036554 010201      MOV      R2,R1          ;SET POINTER TO RK611 BASE
10075 036556 012703 000012      MOV      #10,R3         ;SET COUNT FOR 1ST 10 REGS
10076 036562 012120 1$:      MOV      (R1)+,(R0)+    ;STORE RKCS1 THROUGH RKSPAR
10077 036564 005303      DEC      R3
10078 036566 001375      BNE      1$
10079 036570 062701 000002      ADD      #2,R1          ;BUMP POINTER PAST RKDB
10080 036574 005720      TST      (R0)+         ;BUMP POINTER PAST T.RKDB
10081 036576 012703 000004      MOV      #4,R3         ;SET COUNT FOR LAST 5 REGS
10082 036602 012120 2$:      MOV      (R1)+,(R0)+    ;STORE RKMRI THROUGH RKMRI3
10083 036604 005303      DEC      R3
10084 036606 001375      BNE      2$
10085 036610 012603      MOV      (SP)+,R3      ;RESTORE REGISTERS
10086 036612 012601      MOV      (SP)+,R1
10087 036614 012600      MOV      (SP)+,R0

```

```

10088 036616 000002          RTI          ;RETURN
10089          ;SBTTL BIT COUNTER IN A WORD
10090          ;* THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK
10091          ;* BY THE CALLING ROUTINE. THE NUMBER OF BITS FOUND IN THE WORD
10092          ;* ARE PASSED BACK ON THE STACK.
10093
10094 036620 016637 000002 001256 BITCNT: MOV 2(SF), $TMP16 ;GET WORD WHOSE BITS ARE TO BE COUNTED
10095 036626 010346          MOV R3, -(SP) ;STORE R3
10096 036630 005037 001260          CLR $TMP17 ;CLEAR $TMP16 FOR COUNTING
10097 036634 012703 000021          MOV #17., R3 ;SET A SHIFT COUNTER
10098 036640 000241          CLC ;CLEAR CARRY
10099 036642 006037 001256 1$: ROR $TMP16 ;ROTATE WORD
10100 036646 103407          BCS 3$ ;WAS BIT SHIFTED OUT A 1?
10101 036650 005303          2$: DEC R3 ;NO - DEC COUNT
10102 036652 001373          BNE 1$ ;LOOP IF NOT ZERO
10103 036654 012603          MOV (SP)+, R3 ;RESTORE R3
10104 036656 013766 001260 000002          MOV $TMP17, 2(SP) ;PUT COUNT OF BITS ON STACK
10105 036664 000204          RTS R4 ;RETURN
10106 036666 005237 001260 3$: INC $TMP17 ;BUMP COUNT
10107 036672 000766          BR 2$ ;LOOP
10108
10109          .SBTTL MAINTENANCE CLOCK ROUTINE
10110          ;* THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
10111          ;* ADDRESS AFTER THE CALL. THE FIRST BYTE CONTAINS THE NUMBER
10112          ;* OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
10113          ;* CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
10114          ;* CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
10115
10116 036674          MLOCK:
10117 036674 010046          MOV R0, -(SP) ;; PUSH R0 ON STACK
10118 036676 010146          MOV R1, -(SP) ;; PUSH R1 ON STACK
10119 036700 112400          MOVB (R4)+, R0 ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
10120 036702 112401          MOVB (R4)+, R1 ;GET PARTIAL PHASE ADDRESS COUNT
10121
10122 036704 006300          ASL R0 ;MULTIPLY PHASE ADDRESS COUNT BY 4
10123 036706 006300          ASL R0
10124 036710 060100          ADD R1, R0 ;ADD IN PARTIALS
10125 036712 052762 000400 000026 1$: BIS #MCLK, RKMR1(R2) ;SET CLOCK
10126 036720 042762 000400 000026          BIC #MCLK, RKMR1(R2) ;CLEAR MCLK
10127 036726 005300          DEC R0 ;DECREMENT COUNT
10128 036730 001370          BNE 1$ ;LOOP IF NOT ZERO
10129 036732 012601          MOV (SP)+, R1 ;; POP STACK INTO R1
10130 036734 012600          MOV (SP)+, R0 ;; POP STACK INTO R0
10131 036736 000204          RTS R4
10132          .SBTTL READ AND SORT HEADERS
10133          ;* THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
10134          ;* THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
10135          ;* ASSCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
10136          ;* READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
10137          ;* EACH READ OF THE DATA BUFFER.
10138
10139          CALL: JSR R4, RDSTHD
10140          TCHKOP ;RETURN POINT IF CERR IN READ HDR
10141          ERROR 4 ;OR 5, 6, 7
10142          ERROR 13 ;RETURN IF DATA LATE IN DB UNLOAD
10143          ERROR 2 ;RETURN IF TO SLOW OR

```

```

10144      ;*
10145      ;IF HDR 0 NOT FOUND
10146      036740 104413      RDSTHD: SAVREG
10147      036742 032737 100000 001664      BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
10148      036750 001402      BEQ      20$      ;NO - SKIP
10149      036752 005077 142732      CLR      @KWLADD ;RESET INTERRUPT
10150      036756 012700 000026      20$:     MOV      #26,R0      ;PRESET FOR 26 SECTOR FORMAT
10151      036762 032737 010000 001600      BIT      #CFMT,L.CS1 ;IS 24 SECTOR MODE SET?
10152      036770 001402      BEQ      1$      ;NO - SKIP
10153      036772 012700 000024      MOV      #24,R0      ;ELSE CHANGE TO 24 SECTOR MODE
10154      036776 012701 070414      1$:     MOV      #IBUFF,R1 ;SET POINTER TO INPUT BUFFER
10155      037002 010005      MOV      R0,R5      ;SAVE NUMBER OF SECTORS
10156      037004 010104      MOV      R1,R4      ;SAVE IBUFF ADDRESS
10157      037006 010203      MOV      R2,R3      ;SET UP POINTER TO RKDB
10158      037010 062703 000024      ADD      #RKDB,R3
10159      037014 013762 001626 000010      MOV      DRVNUM,RKCS2(R2) ;LOAD DRIVE NUM
10160      037022 013762 001614 000020      MOV      L.DCYL,RKDCYL(R2) ;LOAD CYLINDER NUM
10161      037030 013762 001606 000006      MOV      L.DA,RKDA(R2) ;LOAD TRACK AND SECTOR
10162
10163      037036 012737 000020 001654      2$:     MOV      #16,TIMCNT ;SET TIME COUNTER
10164      037044 005037 001662      CLR      INTSET ;CLEAR INTERRUPT FLAG
10165      037050 005037 001674      CLR      LCLKTK ;CLEAR TICK COUNTER
10166      037054 053737 001720 001600      BIS      DTYPE,L.CS1 ;SET DRV TYP
10167      037062 013762 001600 000000      MOV      L.CS1,RKCS1(R2) ;LOAD COMMAND
10168
10169      037070 005737 001662      3$:     TST      INTSET ;TEST IF INT OCCURRED
10170      037074 001020      BNE      4$      ;YES - SKIP
10171      037076 004737 035672      JSR      PC,MYTIME ;WAIT 1 MS
10172      037102 005737 001674      TST      LCLKTK ;HAVE WE WAITED 16 MS?
10173      037106 001770      BEQ      3$      ;NO - LOOP ON WAIT
10174
10175      037110 062766 000006 000006      ADD      #6,6(SP) ;SET RETURN FOR TO SLOW
10176      037116 104420      TGETRK ;GET RK REGS
10177      037120 012737 054160 001370      MOV      #EM4,EM2N ;LOAD MESSAGE "TO SLOW/NOT COMPLETE"
10178      037126 012737 050702 001372      MOV      #OPER24,DH2N ;LOAD COMMAND "READ HEADER" FOR REPORT
10179      037134 000466      BR      10$ ;SKIP
10180
10181      037136 005762 000000      4$:     TST      RKCS1(R2) ;TEST FOR CONTROLLER ERROR
10182      037142 100474      BMI      11$ ;YES - SKIP
10183
10184      037144 011324      MOV      (R3),(R4)+ ;STORE HEADERS
10185      037146 011324      MOV      (R3),(R4)+
10186      037150 011324      MOV      (R3),(R4)+
10187
10188      037152 005762 000010      TST      RKCS2(R2) ;TEST IF DATA LATE
10189      037156 100443      BMI      8$ ;YES - SKIP
10190
10191      037160 005300      DEC      R0 ;DEC SECTOR COUNT
10192      037162 001325      BNE      2$ ;IF NOT ZERO - LOOP
10193
10194      037164 032737 100000 001664      BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
10195      037172 001403      BEQ      5$      ;NO - SKIP
10196      037174 012777 000100 142506      MOV      #BIT6,@KWLADD ;SET INTERRUPT ENABLE
10197      037202 032761 000037 000002      5$:     BIT      #37,2(R1) ;HEADER AT TOP OF BUFF=HEAD 0?
10198      037210 001413      BEQ      6$      ;YES - SKIP
10199      037212 012124      MOV      (R1)+,(R4)+ ;ELSE MOV THIS HEADER TO BOTTOM

```

```

10200 037214 012124          MOV      (R1)+,(R4)+
10201 037216 012124          MOV      (R1)+,(R4)+
10202
10203 037220 005305          DEC      R5                ;TEST FO INSURE HEAD 0 IS FOUND
10204 037222 001367          BNE     5$                ;IF ALL HEADERS NOT CHECKED - LOOP
10205 037224 012737 056730 001370  MOV     #EM56,EM2N        ;ELSE "HEADER 0 NOT FOUND" MESSAGE
10206 037232 005037 001372  CLR     DH2N
10207 037236 000421          BR      9$                ;SKIP
10208
10209 037240 012700 070414 6$:    MOV     #IBUFF,RO        ;GET TOP OF IBUFF
10210 037244 012120 7$:    MOV     (R1)+,(RO)+      ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
10211 037246 012120          MOV     (R1)+,(RO)+
10212 037250 012120          MOV     (R1)+,(RO)+
10213 037252 020004          CMP     RO,R4            ;ALL HEADERS MOVED?
10214 037254 001373          BNE     7$                ;NO - LOOP
10215
10216 037256 062766 000010 000006  ADD     #10,6(SP)        ;SET UP FOR GOOD RETURN
10217 037264 000423          BR      11$
10218
10219 037266 012737 055173 001500 8$:    MOV     #EMDLT,EM13N     ;"DATA LATE SET RESULT OF READ DB"
10220 037274 012737 057341 061454  MOV     #EMRDB,DF011A
10221 037302 062766 000004 000006 9$:    ADD     #4,6(SP)        ;SET ERROR RETURN
10222 037310 104420          TGETRK                ;GET RK REGS
10223 037312 013700 001302 10$:   MOV     $TESTN,RO       ;GET TEST NUMBER
10224 037316 006300          ASL     RO              ;SHIFT FOR INDEX
10225 037320 016037 033516 001264  MOV     $$SWO8TB(RO),$ESCAPE ;SET ESCAPE
10226 037326 162737 000002 001264  SUB     #2,$ESCAPE      ;TO NEXT SCOPE CALL
10227
10228 037334 104414          11$:   RESREG
10229 037336 000204          RTS     R4
10230
10231          .SBTTL  GET DRIVE STATUS
10232          ;*      THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
10233          ;*      THROUGH $REG17.  THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
10234          ;*      THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
10235          ;*      IN THE REGISTERS.
10236          ;*
10237          ;*CALL: JSR      R4,GETDRS
10238          ;*      BR      ERROR PROCESSING          ERROR RETURN
10239          ;*      BR      NO ERROR PROCESSING       GOOD RETURN
10240
10241 037340 104413          GETDRS: SAVREG
10242 037342 012762 100000 000000  MOV     #CLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
10243 037350 012700 001202          MOV     #$REG10,RO    ;PRESET ALL STATUS STORAGE TO
10244 037354 012701 000010          MOV     #8,R1         ;ALL ONES
10245 037360 012720 177777 1$:    MOV     #177777,(RO)+
10246 037364 005301          DEC     R1
10247 037366 001374          BNE     1$
10248 037370 012700 001206          MOV     #$REG12,RO    ;SET POINTER TO REG12 FOR A01 & B01
10249 037374 012701 000001          MOV     #1,R1         ;PRESET FOR PAIR ONE.
10250 037400 005037 001230          CLR     $TMP3        ;CLEAR ERROR SWITCH
10251 037404 013762 001610 000010 2$:    MOV     L.CS2,RKCS2(R2);LOAD DRIVE #
10252 037412 010162 000026          MOV     R1,RKMR1(R2) ;LOAD MR1
10253 037416 012762 000001 000000  MOV     #BIT0,RKCS1(R2);DO SELECT
10254 037424 012703 000050          MOV     #40,R3        ;WAIT FOR A FEW MICRO RECORDS TO
10255 037430 005303          3$:    DEC     R3            ;BIT SELECT FINISH

```

10256	037432	001376				BNE	3\$		
10257	037434	032762	100000	000000		BIT	#CERR,RKCS1(R2)	; ANY ERROR SET AS A RESULT OF SELECT?	
10258	037442	001415				BEQ	4\$; NO - SKIP	
10259	037444	032762	024000	000000		BIT	#CTO!SPAR,RKCS1(R2)	; CHECK IF TIMEOUT OR PARITY ERROR	
10260	037452	001004				BNE	8\$; YES - SKIP	
10261	037454	032762	037400	000010		BIT	#37400,RKCS2(R2)	; TEST FOR ERRORS:	
10262								NED!UPE!MDS!UFE!NEM!PGE	
10263	037462	001405				BEQ	4\$; NO - SKIP	
10264	037464	012737	000001	001230	8\$:	MOV	#1,\$TMP3	; SET ERROR FLAG	
10265	037472	022020				CMP	(R0)+,(R0)+	; BUMP TO LET THAT PAIR STAY ALL 1'S.	
10266	037474	000404				BR	5\$; SKIP	
10267	037476	016220	000034		4\$:	MOV	RKMR2(R2),(R0)+	; STORE A WORD	
10268	037502	016220	000036			MOV	RKMR3(R2),(R0)+	; STORE B WORD	
10269	037506	012762	100000	000000	5\$:	MOV	#CCLR,RKCS1(R2)	; CLEAR ANY OLD ERROR IN CONTROLLER	
10270	037514	005701				TST	R1	; IS R1 A 0 (LAST TRANSFER, PAIR 0)	
10271	037516	001410				BEQ	6\$; YES - SKIP	
10272	037520	005201				INC	R1	; ELSE BUMP TO NEXT PAIR	
10273	037522	022701	000004			CMP	#4,R1	; PAIR 3 JUST STORED?	
10274	037526	001326				BNE	2\$; NO - SKIP	
10275	037530	005001				CLR	R1	; ELSE SET TO PAIR 0	
10276	037532	012700	001202			MOV	#\$REG10,R0	; PRESET POINTER FOR PAIR 0	
10277	037536	000722				BR	2\$; GO GET THEM	
10278	037540	104414			6\$:	RESREG		; EXIT HERE	
10279	037542	005737	001230			TST	\$TMP3	; ANY ERROR IN STATUS GETTING	
10280	037546	001001				BNE	7\$; YES - SKIP	
10281	037550	005724				TST	(R4)+	; ELSE BUMP PART ERROR	
10282	037552	000204			7\$:	RTS	R4	; RETURN	
10283									
10284						.SBTTL		SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST	
10285						.*		THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR	
10286						.*		COMMAND. CERR AND DI ARE MONITORED FOR A SHORT	
10287						.*		PERIOD OF TIME DURING WHICH THEY SHOULD BOTH RESET.	
10288						.*			
10289						.*		IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.	
10290						.*			
10291						.*		IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE	
10292						.*		ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE	
10293						.*		RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN	
10294						.*		SKIPS OVER THE ERROR CALL AND TEST ABORT.	
10295						.*			
10296						.*		THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY	
10297						.*		AN ERROR MESSAGE AND BRANCH TO END OF TEST. THIS	
10298						.*		IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO	
10299						.*		THE TEST.	
10300						.*			
10301	037554					SSINIT:			
10302	037554	010046				MOV	R0,-(SP)	; PUSH R0 ON STACK	
10303	037556	010146				MOV	R1,-(SP)	; PUSH R1 ON STACK	
10304	037560	012701	000007			MOV	#7,R1	; SET CLEAR COUNT	
10305	037564	012700	001600			MOV	#L,CS1,R0	; GET ADDRESS OF "L" REGS	
10306	037570	012720	000100			MOV	#100,(R0)+	; PRESET CS1 INTR ENABLE	
10307	037574	005020			7\$:	CLR	(R0)+	; CLEAR THE NEXT	
10308	037576	005301				R1		; COUNT 0?	
10309	037600	001375				BNE	7\$; NO - LOOP	
10310	037602	012762	000040	000010		MOV	#SCLR,RKCS2(R2)	; CLEAR SUBSYSTEM	
10311	037610	012737	000012	001222		MOV	#10,,\$TMP0	; SET A COUNTER	


```

10312 037616 016237 000000 001540 1$: MOV RKCS1(R2),T.CS1 ;GET CS1
10313 037624 032737 140000 001540 BIT #CERR!DI,↑.CS1 ;TEST IF ERROR OR DI SET
10314 037632 001433 BEQ 2$ ;NO - SKIP TO READY TEST
10315 037634 005337 001222 DEC $TMPD ;ELSE DECREMENT COUNTER
10316 037640 001366 BNE 1$ ;AND LOOP
10317 037642 032737 100000 001540 BIT #CERR,T.CS1 ;TEST - IS IT CERR STILL SET
10318 037650 001404 BEQ 3$ ;NO - SKIP TO DI MESSAGE
10319 037652 012737 054253 001400 MOV #EM5,EM3N ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
10320 037660 000403 BR 4$
10321 037662 012737 054317 001400 3$: MOV #EM6,EM3N ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
10322 037670 104420 4$: TGETRK
10323 037672 013700 MOV $TESTN,RO ;GET PRESENT TEST NUMBER
10324 037676 006300 ASL RO ;SHIFT FOR INDEX
10325 037700 016037 033516 001264 MOV $$SWOBTBL(RO),$ESCAPE ;LOAD ESCAPE TO ABORT TEST
10326 037706 162737 000002 001264 SUB #2,$ESCAPE ;SET TO NEXT SCOPE CALL
10327 037714 012601 MOV (SP)+,R1 ;POP STACK INTO R1
10328 037716 012600 MOV (SP)+,RO ;POP STACK INTO RO
10329 037720 000414 BR 6$ ;SKIP TO EXIT
10330 037722 032737 000200 001540 2$: BIT #RDY,T.CS1 ;TEST READY SET
10331 037730 001004 BNE 5$ ;YES - GOOD EXIT
10332 037732 012737 054376 001400 MOV #EM7,EM3N ;MESSAGE (SUBSYS CLR NOT SET READY)
10333 037740 000753 BR 4$
10334 037742 012601 5$: MOV (SP)+,R1 ;RESTORE REGS
10335 037744 012600 MOV (SP)+,RO
10336 037746 062716 000002 ADD #2,(SP) ;GOOD RETURN
10337 037752 000002 6$: RTI
10338
10339 .SBTTL WORD COUNT AT END OF OPERATION CHECK
10340 ;* THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
10341 ;* THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF UNEQUAL, THE
10342 ;* ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
10343 ;*
10344 ;*CALL: JSR R4,CHKWC
10345 ;* .WORD ;EXPECTED WC VALUE
10346
10347 037754 012437 050322 CHKWC: MOV (R4)+,EXPWC ;STORE EXPECTED VALUE
10348
10349 037760 023737 050322 001542 CMP EXPWC,T.WC ;COMPARE
10350 037766 001406 BEQ 1$ ;EQUAL - SKIP
10351 037770 052737 000001 050360 BIS #WCERR,GRP4ER ;SET ERROR FLAG
10352 037776 013737 001542 050336 MOV T.WC,REALWC ;STORE REAL WORD COUNT
10353 040004 000204 1$: RTS R4 ;RETURN.
10354
10355 .SBTTL BUS ADDRESS AT END OF OPERATION CHECK
10356 ;* THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
10357 ;* A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
10358 ;* INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.
10359 ;* IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA
10360 ;* AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)
10361 ;*
10362 ;* IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
10363 ;* ADDRESS IS THE STARTING BUS ADDRESS.
10364 ;*CALL: JSR R4,CHKBA
10365
10366 040006 010046 CHKBA: MOV RO,-(SP)
10367 040010 010146 MOV R1,-(SP)

```

10368	040012	010346			MOV	R3, -(SP)	
10369	040014	032737	000020	001610	BIT	#BA1, L.CS2	; TEST IF BAI SET
10370	040022	001404			BEQ	4\$; NO - SKIP
10371	040024	013737	001604	050326	MOV	L.BA, EXPBA	; STORE EXPECTED BA (SAME AS STARTING BA)
10372	040032	000441			BR	3\$	
10373	040034	013700	001602		4\$: MOV	L.WC, R0	; GET INITIAL WORD COUNT
10374	040040	005400			NEG	R0	
10375	040042	113703	001601		MOVB	L.CS1+1, R3	; GET BA16 & BA17
10376	040046	042703	177774		BIC	#177774, R3	; CLEAR UNWANTED BITS
10377							
10378	040052	005700			TST	R0	; TEST IF INITIAL WORD COUNT 0
10379	040054	001003			BNE	6\$; NO - SKIP
10380	040056	062703	000002		ADD	#2, R3	; ADD 2 TO BA16, 17 (65K WORD XFER)
10381	040062	000407			BR	9\$; SKIP
10382	040064	005700			6\$: TST	R0	; TEST IF INITIAL WC BIT 15 SET
10383	040066	100001			BPL	5\$; NO - SKIP
10384	040070	005203			INC	R3	; BUMP BA16, 17 (32K WORD XFER)
10385	040072	006300			5\$: ASL	R0	; SHIFT WORD COUNT TO MAKE MEM ADD CNT
10386	040074	063700	001604		ADD	L.BA, R0	; ADD IN START BUFFER ADD
10387	040100	005503			ADC	R3	; IF CARRY - ADD INTO BA16, 17
10388	040102	013701	001542		9\$: MOV	T.WC, R1	; GET END OF OPERATION WORD COUNT
10389	040106	001411			BEQ	1\$; BRANCH IF ZERO
10390	040110	005401			NEG	R1	
10391	040112	005701			TST	R1	; TEST END OPERATION WC BIT 15 SET
10392	040114	100001			BPL	7\$; NO - SKIP
10393	040116	005303			DEC	R3	; DEC BA 16, 17 (32K WC LEFT)
10394	040120	006301			7\$: ASL	R1	; SHIFT WC TO MAKE MEM ADD CNT
10395	040122	160100			SUB	R1, R0	; SUB FROM COMPUTED BUS ADDRESS
10396	040124	005603			SBC	R3	; SUB CARRY FROM BA16, 17
10397	040126	010337	050324		MOV	R3, EXPUBA	; STORE EXPECTED UPPER BA BITS
10398	040132	010037	050326		1\$: MOV	R0, EXPBA	
10399	040136	020037	001544		3\$: CMP	R0, T.BA	; EQUAL TO COMPUTED?
10400	040142	001406			BEQ	2\$; YES - SKIP
10401	040144	052737	000004	050360	BIS	#BAERR, GRP4ER	; ELSE SET BAERR FLAG
10402	040152	013737	001544	050342	MOV	T.BA, REALBA	; STORE REAL BUS ADDRESS
10403	040160	113703	001541		2\$: MOVB	T.CS1+1, R3	; GET REAL UPPER BA
10404	040164	042703	177774		BIC	#177774, R3	; CLEAR UNWANTED BITS
10405	040170	020337	050324		CMP	R3, EXPUBA	; CHECK IF EQUAL
10406	040174	001405			BEQ	8\$; YES - SKIP
10407	040176	052737	000002	050360	BIS	#UBAERR, GRP4ER	; ELSE SET UBA ERROR
10408	040204	010337	050340		MOV	R3, REALUB	; STORE REAL UPPER BA
10409	040210	012603			8\$: MOV	(SP)+, R3	
10410	040212	012601			MOV	(SP)+, R1	
10411	040214	012600			MOV	(SP)+, R0	
10412	040216	000204			RTS	R4	

```

10413
10414 .SBTTL CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION
10415 ;* THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA
10416 ;* ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE
10417 ;* OPERATION. THE CONTENTS OF THE LOAD REGISTER STORAGE ARE
10418 ;* COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE
10419 ;* NECESSARY CALCULATION.
10420 ;*
10421 ;* ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES
10422 ;* STORAGE EXPCYL, EXPTRK, EXPSEC. ALL 3 ARE CHECKED AND
10423 ;* IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE
    
```

```

10424      ;*      ERROR FLAGS FIELD (GRP4ER) IS SET.
10425      ;*
10426      ;*CALL: JSR      R4,CHKCTS
10427
10428      040220  104413      CHKCTS: SAVREG
10429      040222  013700  001602      MOV      L.WC,R0      ;GET SPECIFIED WORD COUNT
10430      040226  005400      NEG      R0      ;NEGATE IT
10431      040230  013701  001542      MOV      T.WC,R1      ;GET END OF OPERATION WORD COUNT
10432      040234  001401      BEQ      10$      ;IF ZERO - SKIP
10433      040236  005401      NEG      R1      ;NEGATE IT
10434      040240  160100      10$:     SUB      R1,R0      ;COMPUTE ACTUAL WORDS TRANSFERRED
10435      040242  005001      CLR      R1      ;CLEAR R1 FOR COUNTING
10436      ;
10437      ; THE FOLLOWING CODE DETERMINES HOW MAY SECTORS OF DATA HAS BEEN
10438      ; TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE
10439      ; END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR
10440      ; IS CALCULATED.
10440      040244  022700  000400      1$:     CMP      #400,R0
10441      040250  003004      BGT      2$
10442      040252  005201      INC      R1
10443      040254  162700  000400      SUB      #400,R0
10444      040260  000771      BR       1$
10445      040262  005700      2$:     TST      R0
10446      040264  001401      BEQ      3$
10447      040266  005201      INC      R1
10448      ;
10449      ; AT THIS POINT R1 HAS A COUNT OF THE
10450      ; NUMBER OF FULL SECTOR TRANSFER + 1 IF A
10451      ; PARTIAL SECTOR WAS TRANSFERRED.
10451      040270  012703  000026      3$:     MOV      #26,R3
10452      040274  032737  010000  001600      BIT      #CFMT,L.CS1
10453      040302  001402      BEQ      4$
10454      040304  012703  000024      MOV      #24,R3
10455      ;
10456      ; R3 HAS BEEN SET UP WITH THE NUMBER
10457      ; OF SECTORS IN A TRACK FOR THE FORMAT USED
10457      040310  013737  001614  050330      4$:     MOV      L.DCYL,EXPCYL      ;GET STARTING VALUES FOR CYLINDER
10458      040316  113704  001607      MOVVB   L.DT,R4      ;TRACK
10459      040322  042704  177400      BIC     #177400,R4
10460      040326  113705  001606      MOVVB   L.DS,R5      ;SECTOR
10461      040332  042705  177400      BIC     #177400,R5
10462      040336  005301      DEC     R1      ;ADJUST COUNT FOR ZERO DETECT
10463      040340  005205      5$:     INC     R5      ;BUMP SECTOR COUNT
10464      040342  020503      CMP     R5,R3      ;DID THIS MAKE SECTOR COUTN > 1 TRACK?
10465      040344  001010      BNE    6$      ;NO - SKIP
10466      040346  005005      CLR     R5      ;ELSE CLEAR SECTOR COUNT
10467      040350  005204      INC     R4      ;BUMP TRACK COUNT
10468      040352  022704  000003      CMP     #3,R4      ;DID THIS MAKE TRK COUNT > 1 CYLINDER?
10469      040356  001003      BNE    6$      ;NO - SKIP
10470      040360  005004      CLR     R4      ;ELSE CLEAR TRACK COUNT
10471      040362  005237  050330      INC     EXPCYL      ;BUMP CYLINDER COUNT
10472      040366  005301      6$:     DEC     R1      ;DEC COUNT
10473      040370  001363      BNE    5$      ;IF ZERO - EXIT
10474      040372  010437  050334      MOV     R4,EXPTRK      ;STORE EXPECTED TRACK
10475      040376  010537  050332      MOV     R5,EXPSEC      ;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10476      040402  023737  001560  050330      CMP     T.DCYL,EXPCYL      ;TEST IF CYLINDER OK
10477      040410  001403      BEQ     7$      ;YES - SKIP
10478      040412  052737  000010  050360      BIS     #CYLERR,GRP4ER      ;NO - SET ERROR FLAG
10479      040420  120437  001547      7$:     CMPB   R4,T.DA+1      ;TEST TRACK OK
    
```

```

10480 040424 001403          BEQ      8$          ;YES - SKIP
10481 040426 052737 000020 050360    BIS      #TRKERR,GRP4ER ;NO - SET ERROR FLAG
10482 040434 120537 001546          CMPB    R5,T.DA      ;TEST SECTOR COUNT OK
10483 040440 001403          BEQ      9$          ;YES - SKIP
10484 040442 052737 000040 050360    BIS      #SECERR,GRP4ER ;USE SET ERROR FLAG
10485 040450 012700 050336          MOV     #REALWC,RO
10486 040454 013720 001542          MOV     T.WC,(RO)+   ;STORE REAL WORD COUNT
10487 040460 013720 001544          MOV     T.BA,(RO)+   ;STORE REAL BUS ADDRESS
10488 040464 013720 001560          MOV     T.DCYL,(RO)+ ;STORE REAL CYLINDER ADDRESS
10489 040470 113710 001547          MOV     T.DA+1,(RO) ;STORE REAL TRACK ADDRESS
10490 040474 005720          TST     (RO)+
10491 040476 113710 001546          MOV     T.DA,(RO)   ;STORE REAL SECTOR ADDRESS
10492 040502 104414          RESREG
10493 040504 000204          RTS      R4

10494
10495          .SBTTL  OPERATION CHECK ROUTINE
10496          ;*      THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME SOFTWARE
10497          ;*      ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:
10498          ;*      THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION
10499          ;*      FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH
10500          ;*      THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN
10501          ;*      ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED
10502          ;*      BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL
10503          ;*      BE THE PROPER ITEM FOR THE FORMAT REQUIRED BY THIS
10504          ;*      ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4,5,6,
10505          ;*      7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION
10506          ;*      OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT
10507          ;*      FORMAT.
10508
10509 040506 104413          CHKWE:  SAVREG
10510 040510 011600          MOV     (SP),RO      ;GET POINTER TO ERROR WORDS
10511 040512 012037 001242          MOV     (RO)+,$TMP10 ;STORE EXPECTED ERROR GROUP 1
10512 040516 012037 001244          MOV     (RO)+,$TMP11 ;STORE EXPECTED ERROR GROUP 2
10513 040522 012037 001246          MOV     (RO)+,$TMP12 ;STORE EXPECTED ERROR GROUP 3
10514 040526 010016          MOV     RO,(SP)      ;STORE RETURN
10515 040530 012737 177777 001250          MOV     #-1,$TMP13   ;SET FLAG - EXPECTED ERROR
10516 040536 000403          BR      CHKST

10517
10518 040540 104413          CHKOP:  SAVREG
10519 040542 005037 001250          CLR     $TMP13      ;RESET EXPECTED ERROR FLAG
10520
10521 040546 104420          CHKST:  TGETRK      ;GET 611 REGS IO TRAP
10522 040550 005037 050352          CLR     GRP1ER      ;CLEAR ERROR FLAGS
10523 040554 005037 050354          CLR     GRP2ER
10524 040560 005037 050356          CLR     GRP3ER
10525 040564 005037 050360          CLR     GRP4ER
10526 040570 005037 050474          CLR     GPSUMF      ;CLEAR SUMMARY FLAGS
10527 040574 032737 024000 001540          BIT     #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
10528 040602 001111          BNE    4$           ;YES - SKIP
10529 040604 032737 177400 001550          BIT     #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
10530 040612 001105          BNE    4$           ;YES - SKIP
10531 040614 032737 000070 001552          BIT     #DSERBIT,T.DS  ;TEST IF ERROR SET IN DS
10532 040622 001101          BNE    4$           ;YES - SKIP
10533 040624 005737 001554          TST     T.ER        ;TEST IF ERROR SET IN ER
10534 040630 001076          BNE    4$           ;YES - SKIP
10535 040632 032737 100000 001540          BIT     #CERR,T.CS1  ;COMBINED ERROR SET?

```

```

10536 040640 001405          BEQ      9$          ;NO - SKIP
10537 040642 052737 100000 050352  BIS      #CERNER,GRP1ER ;SET ERROR FLAG IN GROUP 1
10538 040650 000137 041400          JMP      25$          ;SKIP
10539
10540
10541
10542
10543
10544
10545
10546
10547
10548
10549 040654 005737 001250          9$: TST      $TMP13          ;TEST IF ERROR EXPECTED
10550 040660 001402          BEQ      62$          ;NO - SKIP
10551 040662 000137 041400          JMP      25$          ;YES - JUMP
10552 040666 013700 001540          62$: MOV      T,CS1,RO          ;GET CS1
10553 040672 042700 177741          BIC      #177741,RO          ;CHECK IF OPERATION IS READ DATA,
10554 040676 022700 000020          CMP      #20,RO          ;WRITE DATA, OR WRITE CHECK. IF
10555 040702 002445          BLT      3$          ;NOT, SKIP ALL CHECKING IN GROUP
10556 040704 022700 000030          CMP      #30,RO          ;FOUR
10557 040710 003042          BGT      3$
10558 040712 004437 037754          JSR      R4,CHKWC          ;CHECK WORD COUNT
10559 040716 000000          .WORD   0          ;EXPECTED WORD COUNT
10560 040720 004437 040006          JSR      R4,CHKBA          ;CHECK BUS ADDRESS
10561 040724 004437 040220          JSR      R4,CHKCTS          ;CHECK CYL, TRACK, & SECTOR
10562 040730 005737 050360          TST      GRP4ER          ;ANY GROUP 4 ERRORS?
10563 040734 001430          BEQ      3$          ;NO - SKIP
10564 040736 016037 050476 061434  MOV      CMDNLB(RO),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
10565 040744 013700 050360          MOV      GRP4ER,RO          ;PUT GROUP 4 ERROR FLAG IN RO
10566 040750 005001          CLR      R1          ;CLEAR R1 FOR INDEX COUNTER
10567 040752 006200          1$: ASR      RO          ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
10568 040754 103402          BCS      2$          ;BE REPORTED, REST ARE IGNORED
10569 040756 005720          TST      (RO)+          ;WHEN AN ERROR BIT IS FOUND,
10570 040760 000774          BR       1$          ;GET THE ERROR MESSAGE ASSOCIATED
10571 040762 016037 050362 001450  2$: MOV      GRP4MS(RO),EM10N ;WITH IT AND SET ERROR TABLE ITEM TO
10572 040770 016037 050322 001202  MOV      EXPWC(RO),$REG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
10573 040776 016037 050336 001204  MOV      REALWC(RO),$REG11 ;WITH EXPECTED & IS VALUES
10574 041004 104414          RESREG          ;RESTORE REGISTER
10575 041006 012776 000010 000000  MOV      #10,(SP)          ;MAKE THE ERROR CALL POINT TO THE
10576 041014 000002          RTI          ;RIGHT TABLE ENTRY & RETURN.
10577
10578
10579 041016 104414          3$: RESREG
10580 041020 062716 000002          ADD      #2,(SP)          ;BUMP RETURN PAST ERROR
10581 041024 000002          RTI          ;RETURN
10582
10583
10584 041026 012700 050352          4$: MOV      #GRP1ER,RO          ;SET UP GENERAL REGISTER AS POINTER
10585 041032 012701 001540          MOV      #T.CS1,R1          ;CS1
10586 041036 012703 001550          MOV      #T.CS2,R3          ;CS2
10587 041042 012704 001552          MOV      #T.DS,R4          ;DS
10588 041046 012705 001554          MOV      #T.ER,R5          ;AND ER
10589
10590 041052 051510          BIS      (R5),(RO)          ;SET ALL BITS IN GRP1ER THAT
10591

```

CODE TO CHECK WORD COUNT, BUFFER ADDRESS, CYLINDER, TRACK, AND SECTOR AT THE END OF THE OPERATION. THIS IS DONE ONLY IF CERR NOT SET BY THE OPERATION.

ALL OF THE ABOVE CONDITIONS ARE CHECKED AND A BIT SET FOR EACH CHECK THAT FAILS. HOWEVER, ONLY ONE ERROR IS REPORTED. THE ORDER OF PRIORITY FOR REPORTING THE ERROR IS THE ORDER LISTED ABOVE.

THE FOLLOWING CODE BUILDS THE GROUP 1, 2, & 3 ERROR WORDS.

10592	041054	042710	120701		BIC	#ILF!ECH!BSE!HVRC!OPI!DCK,(R0) ; CLEAR ALL THAT DON'T BELONG GRP1
10593						
10594	041060	032711	020000		BIT	#SPAR,(R1) ;TEST IF SPAR SET
10595	041064	001402			BEQ	5\$;NO - SKIP
10596	041066	052710	000001		BIS	#SPARERR,(R0) ;SET SPAR ERROR FLAG
10597						
10598	041072	032714	000010	5\$:	BIT	#ACLO,(R4) ;TEST ACLO SET
10599	041076	001402			BEQ	6\$;NO - SKIP
10600	041100	052710	000100		BIS	#ACLOERR,(R0) ;SET ACLO ERROR FLAG
10601						
10602	041104	032714	000020	6\$:	BIT	#SPDLSS,(R4) ;TEST SPEED LOSS SET
10603	041110	001402			BEQ	7\$;NO - SKIP
10604	041112	052710	000200		BIS	#SPDERR,(R0) ;SET SPEED LOSS ERROR FLAG
10605						
10606	041116	032714	000040	7\$:	BIT	#DROT,(R4) ;TEST IF DROT SET
10607	041122	001402			BEQ	8\$;NO - SKIP
10608	041124	052710	000400		BIS	#DROTERR,(R0) ;SET DROT ERROR FLAG
10609						
10610	041130	032711	100000	8\$:	BIT	#CERR,(R1) ;TEST CERR ITSELF SET
10611	041134	001002			BNE	10\$;YES - SKIP
10612	041136	032710	020000		BIT	#NCERWE,(R0) ;SET NO CERR WITH ERROR FLAG
10613						
10614	041142	012700	050354	10\$:	MOV	#GRP2ER,R0 ;SET POINTER TO GROUP 2 ERROR FLAGS
10615						
10616	041146	032715	000100		BIT	#ECH,(R5) ;TEST IF ECH SET
10617	041152	001402			BEQ	11\$;NO - SKIP
10618	041154	052710	000001		BIS	#ECHERR,(R0) ;SET ECH FLAG
10619						
10620	041160	032715	100000	11\$:	BIT	#DCK,(R5) ;TEST DCK SET
10621	041164	001402			BEQ	12\$;NO - SKIP
10622	041166	052710	000002		BIS	#DCKERR,(R0) ;SET DCK ERROR FLAG.
10623						
10624	041172	032713	040000	12\$:	BIT	#WCE,(R3) ;TEST WRITE CHECK ERROR
10625	041176	001402			BEQ	120\$;NO - SKIP
10626	041200	052710	000004		BIS	#WCKERR,(R0) ;SET WCE BIT
10627	041204	032713	100000	120\$:	BIT	#DLT,(R3) ;TEST DATA LATE
10628	041210	001402			BEQ	13\$;NO - SKIP
10629	041212	052710	000010		BIS	#DLTERR,(R0) ;SET DLT ERROR FLAG
10630						
10631	041216	032715	020000	13\$:	BIT	#OPI,(R5) ;TEST OPI SET
10632	041222	001402			BEQ	14\$;NO - SKIP
10633	041224	052710	000020		BIS	#OPIERR,(R0) ;SET OPI ERROR FLAG
10634						
10635	041230	032715	000400	14\$:	BIT	#HVRC,(R5) ;TEST HVRC SET
10636	041234	001402			BEQ	16\$;NO - SKIP
10637	041236	052710	000040		BIS	#HVCERR,(R0) ;SET HVRC FLAG
10638						
10639	041242	032715	000200	16\$:	BIT	#BSE,(R5) ;TEST BSE ERROR FLAG
10640	041246	001402			BEQ	17\$;NO - SKIP
10641	041250	052710	000100		BIS	#BSERR,(R0) ;SET BSE FLAG
10642						
10643	041254	012700	050356	17\$:	MOV	#GRP3ER,R0 ;SET POINTER TO GROUP 3 FLAGS
10644						
10645	041260	032713	010000		BIT	#NED,(R3) ;TEST NED SET
10646	041264	001402			BEQ	18\$;NO - SKIP
10647	041266	052710	000001		BIS	#NEDERR,(R0) ;SET NED FLAG

10648							
10649	041272	032711	004000	18\$:	BIT	#CTO, (R1)	;TEST CTO SET
10650	041276	001402			BEQ	19\$;NO - SKIP
10651	041300	052710	000002		BIS	#CTOERR, (RO)	;SET CTO FLAG
10652							
10653	041304	032713	000400	19\$:	BIT	#UFE, (R3)	;TEST UFE SET
10654	041310	001402			BEQ	20\$;NO - SKIP
10655	041312	052710	000004		BIS	#UFERR, (RO)	;SET UFE FLAG
10656							
10657	041316	032713	001000	20\$:	BIT	#MDS, (R3)	;TEST MDS SET
10658	041322	001402			BEQ	21\$;NO - SKIP
10659	041324	052710	000010		BIS	#MDSERR, (RO)	;SET MDE FLAG
10660							
10661	041330	032713	002000	21\$:	BIT	#PGE, (R3)	;TEST PGE SET
10662	041334	001402			BEQ	22\$;NO - SKIP
10663	041336	052710	000020		BIS	#PGERR, (RO)	;SET PGE FLAG
10664							
10665	041342	032713	004000	22\$:	BIT	#NEM, (R3)	;TEST NEM SET
10666	041346	001402			BEQ	23\$;NO - SKIP
10667	041350	052710	000040		BIS	#NEMERR, (RO)	;SET NEM FLAG
10668							
10669	041354	032713	020000	23\$:	BIT	#UPE, (R3)	;TEST UPE SET
10670	041360	001402			BEQ	24\$;NO - SKIP
10671	041362	052710	000100		BIS	#UPERR, (RO)	;SET UPE FLAG
10672							
10673	041366	032715	000001	24\$:	BIT	#ILF, (R5)	;TEST ILF SET
10674	041372	001402			BEQ	25\$;NO - SKIP
10675	041374	052710	000200		BIS	#ILFERR, (RO)	;SET ILF FLAG.
10676							
10677				:			
10678				:			
10679				:			
10680				:			
10681				:			
10682				:			
10683				:			
10684				:			
10685				:			
10686				:			
10687				:			
10688	041400	005737	001250	25\$:	TST	\$TMP13	;CHECK IF AN ERROR WAS EXPECTED
10689	041404	001423			BEQ	110\$;NO - SKIP
10690	041406	012704	050352		MOV	#GRP1ER,R4	;GET ADDRESS OF ERROR
10691	041412	012705	001242		MOV	#\$TMP10,R5	;GET ADDRESS OF EXPECTED ERRORS
10692							
10693	041416	011500		26\$:	MOV	(R5),R0	;GET EXPECTED ERROR
10694	041420	011401			MOV	(R4),R1	;GET GROUP ERROR FLAGS
10695	041422	020001			CMP	R0,R1	;ARE THEY EQUAL?
10696	041424	001003			BNE	27\$;NO - SKIP
10697	041426	005000			CLR	R0	;CLEAR EXPECTED ED
10698	041430	005001			CLR	R1	;CLEAR OCCURED
10699	041432	000403			BR	28\$	
10700							
10701	041434	010003		27\$:	MOV	R0,R3	;STORE EXPECTED ERRORS
10702	041436	040100			BIC	R1,R0	;RESET EXPECTED THAT OCCURRED
10703	041440	040301			BIC	R3,R1	;RESET OCCURRED THAT EXPECTED

THE FOLLOWING CODE IS EXECUTED ONLY IF ERRORS WERE EXPECTED.
THE FLAG IN \$TMP13 INDICATES IF
AN ERROR WAS EXPECTED AND THE CONTENTS OF TMP10,
TMP11, & TMP12 SPECIFY WHICH ERRORS. THESE ARE COMPARED AGAINST
THE ERRORS FOUND AND STORED IN GRP1ER, GRP2ER, AND GRP3ER.
THE CONTENTS OF GRP1, 2, & 3 ARE MODIFIED TO INDICATE ERRORS THAT
OCCURRED BUT WERE NOT EXPECTED. THE CONTENTS OF \$TMP10, 11,
& 12 ARE MODIFIED TO INDICATE EXPECTED ERRORS THAT DID NOT
OCCUR. BOTH CONDITIONS CAN EXIST AT THE SAME TIME AND MUST
BE REPORTED.

```

10704 041442 010025          28$:  MOV    RD,(R5)+      ;STORE EXPECTED THAT DID NOT OCCUR
10705 041444 010124          MOV    R1,(R4)+      ;STORE OCCURRED THAT WERE NOT EXPECTED
10706 041446 022705 001250  CMP    #STMP13,R5    ;ALL GROUPS CHECKED.
10707 041452 001361          BNE    26$          ;NO - LOOP
10708
10709          ;
10710          ;
10711          ;
10712          ;
10713          ;
10714          ;
10715          ;
10716          ;
10717          ;
10718          ;
10719          ;
10720          ;
10721          ;
10722          ;
10723          ;
10724          ;
10725          ;
10726          ;
10727          ;
10728          ;
10729          ;
10730          ;
10731          ;
10732          ;
10733          ;
10734          ;
10735 041454 005004          110$: CLR    R4            ;CLEAR COUNTERS
10736 041456 005005          CLR    R5            ;CLEAR COUNTERS
10737 041460 012700 001224  MOV    #STMP1,R0     ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
10738 041464 012701 001226  MOV    #STMP2,R1     ;WHERE ASCIZ ADDRESSES GO
10739 041470 012703 050474  MOV    #GPSUMF,R3    ;POINTERS TO GROUP SUMMARY FLAGS
10740 041474 012710 061414  MOV    #DF007A,(R0)  ;PRESET FOR GRP3 ERR MESSAGE BUILD
10741 041500 012711 001442  MOV    #DH7N,(R1)
10742 041504 013746 050356  MOV    GRP3ER,-(SP)  ;GET GROUP 3 ERRORS, PUT ON STACK
10743 041510 004437 036620  JSR    R4,BITCNT    ;GO COUNT NUMBER AT ERRORS
10744 041514 005716          TST    (SP)          ;ANY ERRORS?
10745 041516 001403          BEQ    29$          ;NO - SKIP
10746 041520 061605          ADD    (SP),R5      ;ADD IN ERROR TOTAL
10747 041522 052713 000004  BIS    #GRP3ST,(R3)  ;SET BIT TO INDICATE GROUP 3 ERROR
10748
10749 041526 005726          29$:  TST    (SP)+        ;CLEAR OFF STACK
10750 041530 005737 001250  TST    STMP13        ;ERROR EXPECTED
10751 041534 001412          BEQ    31$          ;NO - SKIP
10752 041536 013746 001246  MOV    STMP12,-(SP)  ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
10753 041542 004437 036620  JSR    R4,BITCNT    ;COUNT NUMBER OF ERRORS.
10754 041546 005716          TST    (SP)          ;WERE THERE ANY
10755 041550 001403          BEQ    30$          ;NO - SKIP
10756 041552 052713 000040  BIS    #GP3NR,(R3)  ;SET GROUP 3 NOT RECEIVED ERROR FLAG
10757 041556 061604          ADD    (SP),R4      ;ADD COUNT TO TOTAL THESE
10758
10759 041560 005726          30$:  TST    (SP)+        ;CLEAR OFF STACK

```

THE FOLLOWING CODE:

- A. DETERMINES WHICH FORMAT IS TO BE USED
- B. LOADS THE ADDRESSES OF THE ASCIZ TEXT INTO THE SELECTED ERROR TABLE ITEM AND FORMAT FIELD
- C. COUNTS THE NUMBER OF ERRORS THAT MUST BE REPORTED
- D. GETS DRIVE STATUS IF GROUP 1 ERROR.

THE DECISION OF WHICH ERROR IS TO BE USED IS BASED ON THE ERROR GROUP (OR GROUPS) THAT HAVE FLAGS SET. IF ANY BIT IS SET IN GROUP 1, 2, OR 3, GROUP 1 FORMAT (ERROR 4 OR 5) WILL BE USED; ANY SET IN GROUP 2 OR 3, GROUP 2 (ERROR 6) WILL BE USED; AND A FLAG SET IN GROUP 3 ONLY, GROUP 3 (ERROR 7) IS USED.

THE FORMAT TO BE USED IN THE CONTROLLING FACTOR IN HOW THE ERROR TRAP IS CHANGED IN THE MAIN CALL. IF GROUP 1 FORMAT IS USED THE ERROR TRAP WILL BE CHANGED TO ERROR 4 OR 5 (DEPENDING ON AVAILABILITY OF DRIVE STATUS), GROUP 2 FORMAT WILL BE ERROR 6, AND GROUP 3 WILL BE ERROR 7. ONLY THE LOW ORDER BYTE OF THE ERROR TRAP WILL BE ALTERED. THE SP WILL BE POINTING TO THE LOCATION THAT CONTAINS THE ERROR CALL TRAP.

IF THE STATUS IS READ FROM THE DRIVE WITH NO PROBLEM, ERROR 4 IS USED. IF ANY ERROR IS ENCOUNTERED READING STATUS, ERROR 5 IS USED. ERROR 5 INCLUDES A WARNING MESSAGE.


```

10760 041562 013746 050354      31$:  MOV      GRP2ER, -(SP)      ;GET GROUP 2 ERRORS FOR COUNTING
10761 041566 004437 036620      JSR      R4, BITCNT          ;COUNT BITS
10762 041572 005716              TST      (SP)                ;ANY SET?
10763 041574 001407              BEQ      32$                 ;NO - SKIP
10764 041576 052713 000002      BIS      #GRP2ST, (R3)       ;SET FLAG FOR GROUP 2 ERRORS
10765 041602 061605              ADD      (SP), R5            ;ADD INTO TOTAL
10766 041604 012710 061370      MOV      #DF006A, (R0)       ;STORE ADDRESS FOR BUILDING REPORT
10767 041610 012711 001432      MOV      #DH6N, (R1)
10768
10769 041614 005726      32$:  TST      (SP)+             ;CLEAR OFF STACK
10770 041616 005737 001250      TST      $TMP13              ;ANY EXPECTED ERRORS
10771 041622 001416              BEQ      34$                 ;NO - SKIP
10772 041624 013746 001244      MOV      $TMP11, -(SP)       ;GET GROUP 2 NOT RECEIVED ERRORS
10773 041630 004437 036620      JSR      R4, BITCNT          ;COUNT NUMBER OF BITS
10774 041634 005716              TST      (SP)                ;ANY SET?
10775 041636 001407              BEQ      33$                 ;NO - SKIP
10776 041640 052713 000020      BIS      #GP2NR, (R3)        ;SET FLAG FOR GROUP 2 NOT RECEIVED
10777 041644 061604              ADD      (SP), R4            ;ADD INTO TOTAL
10778 041646 012710 061370      MOV      #DF006A, (R0)       ;STORE ADDRESS FOR BUILDING REPORT
10779 041652 012711 001432      MOV      #DH6N, (R1)
10780
10781 041656 005726      33$:  TST      (SP)+             ;CLEAR OFF STACK
10782 041660 013746 050352      34$:  MOV      GRP1ER, -(SP)     ;GET GROUP 1 ERROR FLAGS
10783 041664 004437 036620      JSR      R4, BITCNT          ;COUNT THE NUMBER OF BITS
10784 041670 005716              TST      (SP)                ;ANY SET?
10785 041672 001407              BEQ      35$                 ;NO - SKIP
10786 041674 052713 000001      BIS      #GRP1ST, (R3)       ;SET FLAG FOR GROUP 1 ERRORS SET
10787 041700 061605              ADD      (SP), R5            ;ADD INTO TOTAL
10788 041702 012710 061304      MOV      #DF004A, (R0)       ;LOAD ADDRESS FOR BUILDING REPORT
10789 041706 012711 001412      MOV      #DH4N, (R1)
10790 041712 005726      35$:  TST      (SP)+             ;CLEAR OFF STACK
10791 041714 005737 001250      TST      $TMP13              ;ANY EXPECTED ERRORS?
10792 041720 001416              BEQ      60$                 ;NO - SKIP
10793 041722 013746 001242      MOV      $TMP10, -(SP)       ;GET GROUP 1 NO RECEIVED ERROR
10794 041726 004437 036620      JSR      R4, BITCNT          ;COUNT # OF BITS
10795 041732 005716              TST      (SP)                ;ANY SET?
10796 041734 001407              BEQ      36$                 ;NO - SKIP
10797 041736 052713 000010      BIS      #GP1NR, (R3)        ;SET FLAG FOR GROUP 1 NOT RECEIVED
10798 041742 061604              ADD      (SP), R4            ;ADD INTO TOTAL
10799 041744 012710 061304      MOV      #DF004A, (R0)       ;LOAD ADDRESS FOR BUILDING REPORT
10800 041750 012711 001412      MOV      #DH4N, (R1)
10801 041754 005726      36$:  TST      (SP)+             ;CLEAR OFF STACK.
10802 041756 032713 000011      60$:  BIT      #GRP1ST!GP1NR, (R3) ;ANY GROUP 1 ERROR
10803 041762 001414              BEQ      52$                 ;NO - SKIP
10804 041764 042713 040000      BIC      #DRSTER, (R3)
10805 041770 004437 037340      JSR      R4, GETDRS
10806 041774 000401              BR       51$                 ;ERROR RETURN
10807 041776 000406              BR       52$                 ;NO ERROR RETURN
10808 042000 012710 061334      51$:  MOV      #DF005A, (R0)       ;CHANGE TO FORMAT 5 - STORE ADDRESS
10809 042004 012711 001422      MOV      #DH5N, (R1)         ;FOR BUILDING REPORT
10810 042010 052713 040000      BIS      #DRSTER, (R3)       ;SET DRIVE STATUS ERROR
10811 042014
10812
10813      ;      THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS
10814      ;      ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED.
10815      ;      THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE

```

```

10816
10817
10818
10819
10820
10821
10822
10823
10824
10825
10826
10827
10828
10829
10830
10831
10832
10833
10834
10835
10836
10837
10838
10839 042014 032777 020000 137116
10840 042022 001402
10841 042024 000137 042412
10842 042030 005737 001250
10843 042034 001004
10844
10845
10846
10847 042036 012771 060307 000000
10848 042044 000411
10849 042046 012771 060130 000000
10850 042054 032713 000070
10851 042060 001003
10852 042062 012771 060225 000000
10853 042070 013701 001540
10854 042074 042701 177741
10855
10856 042100 016170 050476 000000
10857
10858 042106 032713 000007
10859 042112 001462
10860
10861
10862
10863 042114 013701 050356
10864 042120 012700 050376
10865 042124 005037 001252
10866 042130 012737 000021 001254
10867 042136 000241
10868 042140 006001
10869 042142 103406
10870 042144 062700 000002
10871 042150 005337 001254
    
```

```

PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE
AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER
OF TABLE.

AT THIS TIME
R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
NOT EXPECTED OR
THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
DID NOT OCCUR.
$TMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
$TMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
$TMP12 CONTAINS GROUP 3 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
GRP1ER CONTAINS GROUP 1 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
GRP2ER CONTAINS GROUP 2 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
GRP3ER CONTAINS GROUP 3 ERRORS THAT OCCURRED OR OCCURRED AND WERE NOT EXPECT
(R1)=$TMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
DF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
(RO)=$TMP1 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
THE ERROR ITEM TABLE THAT MUST BE ALTERED TO PROVIDE A
PROPER MESSAGE TO REPORT.
(R3)=$GRSUMF CONTAIN FLAGS TO INDICATE WHICH OF THE GROUP
ERROR FLAG FIELDS HAVE ERROR BITS STORED.
BIT #SW13, $SWR ; IS REPORT INHIBITED?
BEQ 37$ ; NO - SKIP
JMP 49$ ; ELSE EXIT
TST $TMP13 ; WERE ERRORS EXPECTED?
BNE 38$ ; YES - SKIP

; IF NO ERRORS WERE EXPECTED, $TMP10,11, &12 ARE NOT MEANINGFUL
MOV #DH007, $($R1) ; HEADER = ERROR IN OPERATION
BR 39$
MOV #DH005, $($R1) ; PRESET HDRMSG = EXPECTED NOT SET
BIT #GP1NR!GP2NR!GP3NR, (R3) ; ANY NOT RECEIVED ERRORS?
BNE 39$ ; YES - SKIP
MOV #DH006, $($R1) ; SET MESSAGE TO UNEXPECTED ERROR SET
MOV T.CS1, R1 ; GET CS1
BIC #177741, R1 ; CLEAR ALL BUT COMMAND

MOV CMNDLB(R1), $($RO) ; MOVE ADDRESS OF COMMAND MESSAGE
; INTO REPORT
BIT #GRP1ST!GRP2ST!GRP3ST, (R3) ; ANY GPR ERRORS?
BEQ 46$ ; NO - SKIP GPR REPORT

; PRINT ALL THE ERRORS CONTAINED IN THE GPR1,2,3ER(UNEXPECTED ERRORS)
MOV GRP3ER, R1 ; GET GROUP 3 ERROR FLAGS
MOV #GRP3MS, RO ; SET POINTER TO GRP3 ERROR MESSAGES
CLR $TMP14 ; CLEAR GROUP PRINTING INDICATOR
MOV #17., $TMP15 ; PRESET SHIFT COUNT
CLC ; CLEAR CARRY
ROR R1 ; ROTATE ERROR FLAGS
BCS 42$ ; WAS BIT SHIFTED OUT SET?
ADD #2, RO ; NO - BUMP POINTER
DEC $TMP15 ; DEC SHIFT COUNT
    
```

```

10872 042154 001371          BNE 41$           ;LOOP IF SHIFT NOT ZERO
10873 042156 000411          BR 44$
10874 042160 011037 042172 42$: MOV (R0),43$      ;GET ERROR MESSAGE ADDRESS FROM TABLE
10875 042164 104401 001273  TYPE , $CRLF      ;TYPE CRLF
10876 042170 104401          TYPE              ;TYPE ERROR MESSAGE
10877 042172 000000          .WORD            ;MESSAGE ADDRESS GOES HERE
10878 042174 005305          DEC R5            ;DECEREMENT TOTAL ERROR COUNT.
10879 042176 001362          BNE 141$         ;LOOP IF ZERO
10880 042200 000427          BR 46$           ;ELSE EXIT GPR ERROR PRINT LOOP
10881
10882 042202 005713          44$: TST (R3)      ;TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
10883 042204 100455          BMI 47$         ;YES - SKIP
10884 042206 005737 001252  TST $TMP14      ;PRINTING GROUP 3?
10885 042212 001007          BNE 45$         ;NO -SKIP
10886 042214 013701 050354  MOV GRP2ER,R1   ;ELSE SET TO GROUP 2, GET GRP2ER
10887 042220 012700 050416  MOV #GRP2MS,R0 ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10888 042224 005237 001252  INC $TMP14      ;BUMP TO INDICATE PRINTING GROUP 2
10889 042230 000737          BR 40$         ;GO RESTART PRINT LOOP
10890 042232 022737 000002 001252 45$: CMP #2,$TMP14   ;PRINTING GROUP 1?
10891 042240 001407          BEQ 46$         ;YES - EXIT GPR ERROR PRINT LOOP.
10892 042242 013701 050352  MOV GRP1ER,R1   ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10893 042246 012700 050434  MOV #GRP1MS,R0 ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10894 042252 005237 001252  INC $TMP14      ;BUMP TO INDICATE PRINTING GROUP 1
10895 042256 000724          BR 40$         ;RESTART PRINT LOOP.
10896
10897 042260 005737 001250 46$: TST $TMP13      ;EXPECTING ERRORS?
10898 042264 001452          BEQ 49$         ;NO - SKIP
10899
10900 ; PRINT ALL ERRORS CONTAINED IN $TMP10, 11, 12(NOT RECEIVED ERRORS)
10901
10902 042266 005713          TST (R3)        ;TEST IF PRINTING NOT RECEIVED ERRORS
10903 042270 100423          BMI 47$         ;YES - SKIP
10904 042272 032713 000070  BIT #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
10905 042276 001445          BEQ 49$         ;NO - SKIP
10906 042300 032713 000007  BIT #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
10907 042304 001404          BEQ 146$       ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
10908 042306 104401 001273  TYPE , $CRLF      ;TYPE CRLF
10909 042312 104401 060225  TYPE , DH006     ;TYPE HEADER FOR PREVIOUS ERRORS
10910 042316 052737 100000 050474 146$: BIS #REPNR,GPSUMF ;SET PRINTING NOT RECEIVED ERRORS SWITCH
10911 042324 010405          MOV R4,R5       ;MOVE TOTAL ERRORS TO R5
10912 042326 013701 001246  MOV $TMP12,R1   ;GET GRP3 NOT RECEIVED ERRORS
10913 042332 012700 050376  MOV #GRP3MS,R0 ;SET POINTER TO GROUP 3 MESSAGES
10914 042336 000672          BR 140$        ;GO START PRINT LOOP
10915 042340 005737 001252 47$: TST $TMP14      ;PRINTING GROUP 3?
10916 042344 001007          BNE 48$         ;NO - SKIP
10917 042346 013701 001244  MOV $TMP11,R1   ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10918 042352 012700 050416  MOV #GRP2MS,R0 ;& SET POINTER TO GROUP 2 MESSAGE TABLE
10919 042356 005237 001252  INC $TMP14      ;BUMP TO INDICATE GROUP 2 PRINTING
10920 042362 000662          BR 40$         ;GO START PRINT LOOP
10921 042364 022737 000002 001252 48$: CMP #2,$TMP14   ;PRINTING GROUP 1?
10922 042372 001407          BEQ 49$         ;YES - EXIT LOOP
10923 042374 013701 001242  MOV $TMP10,R1   ;SET POINTER TO GROUP 1 MESSAGE
10924 042400 012700 050434  MOV #GRP1MS,R0 ;TABLE AND GET GROUP 1 ERRORS.
10925 042404 005237 001252  INC $TMP14      ;BUMP TO INDICATE GROUP 1 PRINTING
10926 042410 000647          BR 40$         ;START LOOP AGAIN.
10927

```

```

10928 042412 032713 000077      49$: BIT #77,(R3) ;TEST IF ANY ERRORS TO BE REPORTED
10929                                     ; GRP1ST!GRP2ST!GRP3ST
10930                                     ; GP1NR!GP2NR!GP3NR
10931 042416 001004             BNE 61$ ;YES - SKIP
10932 042420 104414             RESREG ;ELSE EXIT
10933 042422 062716 000002     ADD #2,(SP) ;BUMP FOR GOOD RETURN
10934 042426 000002             RTI
10935
10936 042430 112776 000007 000000 61$: MOVB #7,a(SP) ;PRESET FOR GROUP 3 ERROR RETURN.
10937 042436 032713 000022     BIT #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10938 042442 001403             BEQ 50$ ;NO - SKIP
10939 042444 112776 000006 000000 MOVB #6,a(SP) ;ELSE SET FOR GROUP 2 ERROR RETURN
10940
10941 042452 032713 000011      50$: BIT #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10942 042456 001411             BEQ 53$ ;NO - SKIP
10943 042460 112776 000004 000000 MOVB #4,a(SP) ;ELSE SET FOR GROUP 1 ERROR RETURN.
10944 042466 032713 040000     BIT #DRSTER,(R3) ;CHECK IF ERROR GETTING DRIVE STATUS
10945 042472 001403             BEQ 53$ ;NO - SKIP
10946 042474 112776 000005 000000 MOVB #5,a(SP) ;ELSE CHANGE RETURN FORM GROUP 1
10947
10948 042502 005737 001264      53$: TST $ESCAPE ;CHECK IF ESCAPE ALREADY SET
10949 042506 001011             BNE 54$ ;YES - SKIP
10950 042510 013700 001302     MOV $TESTN,RO ;SET UP $ESCAPE TO FORCE
10951 042514 006300             ASL RO ;ABORT TO PRESENT TEST AFTER
10952 042516 016037 033516 001264 MOV $SWO8TB(RO),$ESCAPE ;ERROR IS REPORTED
10953 042524 162737 000002 001264 SUB #2,$ESCAPE ;BUT GO TO NEXT SCOPE STATEMENT
10954 042532 104414             54$: RESREG
10955 042534 000002             RTI ;RETURN
10956
10957 ;*****
10958 ;SBTTL BAD SECTOR CHECK
10959 ; THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
10960 ; CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
10961 ; THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
10962 ; L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
10963 ; THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
10964 ; IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
10965 ; IS PLACED ON THE STACK AFTER THE ENTIRE
10966 ; FIELD IS SEARCHED.
10967 ;
10968 ; CALL: JSR R4,BDSRCK
10969 ; <ADDRESS OF FIELD TO BE SEARCHED>
10970 ;*****
10971
10972 042536 012637 001236      BDSRCK: MOV (SP)+,$TMP6 ;STORE OLD R4 CONTENTS
10973 042542 010437 001240     MOV R4,$TMP7 ;GET RETURN ADDRESS
10974 042546 011404             MOV (R4),R4 ;GET POINTER TO FIELD TO BE CHECKED
10975 042550 005037 001234     CLR $TMP5 ;CLEAR A COUNTER
10976 042554 005714             1$: TST (R4) ;TEST IF FIELD HAS NO (OR NO MORE) ENTRIES
10977 042556 100417             BMI 4$ ;YES - EXIT
10978 042560 023724 001614     CMP L.DCYL,(R4)+ ;IS THIS ENTRY FOR THIS CYLINDER?
10979 042564 001012             BNE 3$ ;NO - SKIP
10980 042566 005204             INC R4 ;BUMP TO TRACK
10981 042570 123714 001607     CMPB L.DT,(R4) ;IS ENTRY FOR THIS TRACK?
10982 042574 001005             BNE 2$ ;NO - SKIP
10983 042576 005046             CLR -(SP) ;CLEAR STACK LOCATION

```

10984 042600 114416
10985 042602 005237 001234
10986 042606 000401
10987
10988 042610 005304
10989 042612 005724
10990 042614 000757
10991
10992 042616 013746 001234
10993 042622 013746 001236
10994 042626 013704 001240
10995 042632 005724
10996 042634 000204
10997
10998
10999
11000
11001
11002
11003
11004
11005
11006
11007
11008
11009
11010
11011
11012
11013
11014
11015
11016
11017
11018
11019
11020
11021
11022
11023
11024
11025
11026
11027
11028
11029
11030
11031
11032
11033
11034
11035
11036
11037
11038
11039

```

MOV  B -(R4), (SP) ; PUT SECTOR NUMBER ON STACK
INC  $TMP5 ; BUMP COUNTER
BR   $ ; BRANCH

2$:  DEC  R4 ; DECREMENT POINTER TO WORD ALIGN
3$:  TST  (R4)+ ; BUMP TO NEXT ENTRY
     BR   1$ ; TEST NEXT ENTRY

4$:  MOV  $TMP5, -(SP) ; PUT COUNT ON STACK
     MOV  $TMP6, -(SP) ; PUT OLD R4 CONTENTS BACK ON STACK
     MOV  $TMP7, R4 ; SET UP RETURN
     TST  (R4)+ ; BUMP PAST PARAMETER
     RTS  R4 ; RETURN
    
```

SBTTL DATA GENERATION AND COMPARE ROUTINE

```

CALLS:  JSR  R4, GENCOM
        CONTROL WORD

        JSR  R4, GENCOM
        CONTROL WORD
        LENGTH

        JSR  R4, GENCOM
        CONTROL WORD
        RELOCATION CONSTANT
        LENGTH
    
```

RETURN: RTS R4

R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
CONTROL WORD. THIS IS UNCONDITIONAL.

THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.

THE THIRD CALL IS IDENTICAL TO THE SECOND.

DEFINITION OF CONTROL WORD:

```

BIT 15 - DO COMPARE OPERATION OF Ibuff (SOURCE) TO Obuff
        (DESTINATION). EXPECTED VALUES ARE IN Obuff (DESTINATION).
BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
BIT 10 - CLEAR Ibuff TO PATTERN SELECTED.
BIT 9  - BUILD HEADERS, CONSIDERING BS FILES
BIT 8  - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
BIT 7  - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
            NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
            OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.
    
```

EXPLANATION OF CALLS:

11040
 11041
 11042
 11043
 11044
 11045
 11046
 11047
 11048
 11049
 11050
 11051
 11052
 11053
 11054
 11055
 11056
 11057
 11058
 11059
 11060
 11061
 11062
 11063
 11064
 11065
 11066
 11067
 11068
 11069
 11070
 11071
 11072
 11073
 11074
 11075
 11076
 11077
 11078
 11079
 11080
 11081
 11082
 11083

* THE CALL WITH CONTROL WORD THE ONLY PARAMETER IS USED FOR
 * BUILDING OR COMPARING HEADERS OR RESUMING A COMPARE OPERATION.
 *
 * THE CALL WITH CONTROL WORD AND LENGTH AS PARAMETERS IS USED
 * FOR DATA GENERATION OR COMPARE AND FOR Ibuff INITIALIZATION.
 *
 * THE CALL WITH CONTROL WORD, RELOCATION CONSTANT, AND LENGTH IS
 * USED FOR DATA GENERATION OR COMPARE WITH MEMORY MANAGEMENT.

DESCRIPTION:

THIS ROUTINE IS MULTI-PURPOSE AND WILL PERFORM THE FOLLOWING:
 A. BUILD HEADERS, EITHER 20 OR 22 SECTORS/TRACK MODE. THE ROUTINE WILL BUILD THE HEADERS AS ALL GOOD SECTORS (BIT 8) OR TAKE THE BAD SECTOR FILES (HARDWARE OR SOFTWARE) FOR EITHER FORMAT) INTO ACCOUNT AND BUILD THE HEADERS WITH THE SECTORS MARKED BAD IF ANY SECTORS FOR THE CYLINDER - TRACK ARE LISTED THEREIN (BIT 9).
 B. COMPARE THE CONTENTS OF Ibuff AND Obuff (BIT 15). THE CONTENTS OF THE BUFFER MAY BE HEADERS OR DATA. A HEADER COMPARE OPERATION MAY BE SPECIFIED (BIT 7) WHICH WILL CAUSE THE COMPARE TO BE LIMITED TO 74(8) OR 102(8) WORDS OF HEADERS. THE LENGTH DEPENDS ON THE FORMAT BIT THAT WAS LAST SPECIFIED IN L.CSI. THE HEADERS MAY BE BUILT BEFORE THE COMPARE AS PART OF THE OPERATION (BIT 15 AND BIT 8 OR 9). DATA CAN ALSO BE GENERATED BEFORE THE COMPARE (NON-ZERO BITS 6-0).
 C. RESUME COMPARE OPERATION. IF A COMPARE OPERATION DETECTS A MISCOMPARE, THE ROUTINE RETURNS TO CALLER BUT STORES PARAMETERS SUCH THAT THE COMPARE CAN BE RESUMED. THIS IS DONE BY CALLING GENCOM WITH BIT 14 SET IN THE CONTROL WORD.
 D. DATA GENERATION OR COMPARE USING MEMORY MANAGEMENT. MEMORY MANAGEMENT CAN BE INVOKED FOR EITHER SOURCE OR DESTINATION BUT NOT FOR BOTH. IN THIS MANNER, DATA GENERATION CAN BE MADE TO PLACE DATA ANYWHERE IN AVAILABLE MEMORY. LIKEWISE DATA COMPARE WILL COMPARE THE CONTENTS OF Ibuff TO ANY AREA OF AVAILABLE MEMORY.

GENCOM:

11084	042636				MOV	R0,-(SP)	;;	PUSH R0 ON STACK
11085	042636	010046			MOV	R1,-(SP)	;;	PUSH R1 ON STACK
11086	042640	010146			MOV	R3,-(SP)	;;	PUSH R3 ON STACK
11087	042642	010346			MOV	R5,-(SP)	;;	PUSH R5 ON STACK
11088	042644	010546			MOV	(R4)+,R0	;	GET PARAMETER WORD
11089	042646	012400			MOV	#EM54,EM15N	;	PRESET FOR HEADER COMPARE ERROR
11090	042650	012737	056654	001520	BIT	#BIT7,R0	;	HEADER OPERATION SPECIFIED?
11091	042656	032700	000200		BNE	18\$;	YES - SKIP
11092	042662	001005			MOV	#EM55,EM15N	;	CHANGE FOR DATA COMPARE ERROR
11093	042664	012737	056703	001520	JMP	17\$;	ELSE JUMP TO DATA ROUTINE
11094	042672	000137	043424					
11095	042676							

18\$:

11096	042676	010446				MOV	R4, -(SP)	:: PUSH R4 ON STACK
11097	042700	032700	001400			BIT	#BIT8:BIT9, R0	:: MUST HEADERS BE BUILT?
11098	042704	001002				BNE	19\$:: YES - SKIP
11099	042706	000137	043206			JMP	11\$:: ELSE JUMP TO HEADER COMPARE
11100	042712	113701	001607		19\$:	MOVB	L.DT, R1	:: START HEADER BUILD ROUTINE
11101	042716	013703	001614			MOV	L.DCYL, R3	:: GET TRACK AND CYL
11102	042722	012705	000005			MOV	#5, R5	:: SET COUNT TO SHIFT TRACK FOR HDR WORD
11103								
11104	042726	006301			1\$:	ASL	R1	:: SHIFT TRACK
11105	042730	005305				DEC	R5	:: DECREMENT TRACK
11106	042732	001375				BNE	1\$:: LOOP UNTIL COUNT 0
11107								
11108	042734	012704	000026			MOV	#26, R4	:: PRESET FOR 26 SECTOR MODE
11109	042740	032737	010000	001600		BIT	#CFMT, L.CS1	:: IS IT 24 SECTOR MODE?
11110	042746	001404				BEQ	2\$:: NO - SKIP
11111	042750	012704	000024			MOV	#24, R4	:: CHANGE COUNT FOR 24 SECTOR MODE
11112	042754	052701	001000			BIS	#BIT9, R1	:: SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
11113								
11114	042760	052701	140000		2\$:	BIS	#BIT15:BIT14, R1	:: SET BS BITS TO INDICATE GOOD SECTOR
11115	042764	012705	072414			MOV	#OBUF, R5	:: SET POINTER TO ADDRESS WHERE HEADERS GO
11116	042770	010325			3\$:	MOV	R3, (R5)+	:: INSERT CYLINDER
11117	042772	010125				MOV	R1, (R5)+	:: INSERT TRACK AND SECTOR
11118	042774	010337	001224			MOV	R3, \$TMP1	:: CALCULATE HVRC WORD
11119	043000	010115				MOV	R1, (R5)	
11120	043002	040137	001224			BIC	R1, \$TMP1	
11121	043006	040315				BIC	R3, (R5)	
11122	043010	053725	001224			BIS	\$TMP1, (R5)+	:: COMPLETE HVRC WORD INSERTION
11123								
11124	043014	005304				DEC	R4	:: DECREMENT HEADER COUNT
11125	043016	001402				BEQ	4\$:: DONE? - YES, SKIP
11126	043020	005201				INC	R1	:: BUMP SECTOR
11127	043022	000762				BR	3\$:: LOOP
11128								
11129	043024	032700	001000		4\$:	BIT	#BIT9, R0	:: MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
11130	043030	001003				BNE	5\$:: YES - SKIP
11131	043032	005700			10\$:	TST	R0	:: IS THIS A COMPARE OPERATION?
11132	043034	100464				BMI	11\$:: YES-GO DO HDR COMPARE
11133	043036	000534				BR	50\$:: ELSE GET OUT
11134								
11135	043040	013737	001640	043076	5\$:	MOV	BSF26P, 6\$:: PRESET FOR BS FACTORY LIST
11136	043046	012737	100000	001224		MOV	#BIT15, \$TMP1	:: SET BIT TO BE RESET IN BAD HEADER
11137	043054	032737	010000	001600		BIT	#CFMT, L.CS1	:: IS THIS 26 SECTOR MODE?
11138	043062	001403				BEQ	8\$:: YES - SKIP
11139	043064	013737	001636	043076		MOV	BSF24P, 6\$:: ELSE CHANGE FOR 24 SECTOR MODE
11140								
11141	043072	004437	042536		8\$:	JSR	R4, BDSRCK	:: GO CHECK FOR BAD SECTOR THIS ADDRESS
11142	043076	000000			6\$:	.WORD	0	:: POINTER TO FILE TO BE CHECKED GOES HERE
11143	043100	012605				MOV	(SP)+, R5	:: GET # OF BAD SECTORS THIS PACK ADDRESS
11144	043102	001417				BEQ	9\$:: SKIP IF ZERO
11145								
11146	043104	011601			7\$:	MOV	(SP), R1	:: GET 1ST BAD SECTOR NUMBER
11147	043106	006301				ASL	R1	:: MULTIPLY SECTOR NUMBER BY 6 TO
11148	043110	006301				ASL	R1	:: LOCATE SECTOR TO BE MARKED BAD
11149	043112	061601				ADD	(SP), R1	
11150	043114	062601				ADD	(SP)+, R1	
11151	043116	062701	000002			ADD	#2, R1	:: ADD 2 FOR 2ND WORD THAT SECTOR

```

11152 043122 043761 001224 072414      BIC      $TMP1,0BUFF(R1) ; CLEAR BIT FOR BAD SECTOR IN HDR
11153 043130 043761 001224 072416      BIC      $TMP1,0BUFF+2(R1) ; CORRECT THE HVRC BIT
11154 043136 005305                DEC      R5 ; DECREMENT BAD SECTOR COUNT
11155 043140 001361                BNE      7$ ; LOOP IF NOT ZERO
11156
11157 043142 032737 100000 001224 9$:      BIT      #BIT15,$TMP1 ; WERE WE DOING BS FACTORY LIST?
11158 043150 001730                BEQ      10$ ; NO - GO CHECK IF COMPARE MUST BE DONE
11159 043152 012737 040000 001224      MOV      #BIT14,$TMP1 ; ELSE SET BIT TO BE RESET IN BAD HDR
11160 043160 013737 001644 043076      MOV      BSS26P,6$ ; PRESET POINTER FOR 26 SECTOR MODE
11161 043166 032737 010000 001600      BIT      #CFMT,L.CS1 ; TEST IF WE ARE DOING 26 SECTOR MODE
11162 043174 001736                BEQ      8$ ; YES - SKIP TO START CHECK
11163 043176 013737 001642 043076      MOV      BSS24P,6$ ; CHANGE POINTER TOR 24 SECTOR MODE
11164 043204 000732                BR       8$ ; SKIP TO START CHECK
11165
11166
11167 043206 012701 000102 001600 11$:     MOV      #102,R1 ; PRESET FOR 102 WORDS OF HEADER
11168 043212 032737 010000                BIT      #CFMT,L.CS1 ; CHECK IF 26 SECTOR MODE
11169 043220 001402                BEQ      12$ ; YES - SKIP
11170 043222 012701 000074                MOV      #74,R1 ; CHANGE TO 74 WORDS OF HEADER
11171
11172 043226 012704 070414 12$:     MOV      #IBUFF,R4 ; SET START OF HEADERS TO BE COMPARED
11173 043232 012705 072414                MOV      #0BUFF,R5 ; SET START OF GOOD HEADERS
11174 043236 005003                CLR      R3 ; CLEAR COUNTER
11175 043240 032700 040000                BIT      #BIT14,RO ; IS THIS A CONTINUATION OF EARLIER COMPARE
11176 043244 001412                BEQ      13$ ; NO - SKIP
11177 043246 013705 001700 28$:     MOV      DESHLD,R5 ; GET VALUES WHERE PREVIOUS CHECK STOPPED
11178 043252 013704 001702                MOV      SRCHLD,R4 ; DESTINATION AND SOURCE
11179 043256 013703 001704                MOV      WRDNUM,R3 ; WORD NUMBER IN ERROR
11180 043262 013701 001706                MOV      WRDCNT,R1 ; WORD COUNT LEFT IN COMPARE
11181 043266 005701                TST      R1 ; TEST IF WORD COUNT LEFT = 0
11182 043270 001417                BEQ      50$ ; YES - EXIT
11183
11184 043272 032700 030000 13$:     BIT      #BIT12!BIT13,RO ; MEM MANAGE REQUIRED?
11185 043276 001402                BEQ      25$ ; NO - SKIP
11186 043300 005237 177572                INC      @#SRO ; TURN IT ON
11187 043304 022425 25$:     CMP      (R4)+,(R5)+ ; COMPARE THE WORDS
11188 043306 001012                BNE      14$ ; SKIP IF NOT EQUAL
11189 043310 005203                INC      R3 ; BUMP WORD NUMBER IN ERROR
11190 043312 005301                DEC      R1 ; DEC WORD COUNT LEFT IN COMPARE
11191 043314 001373                BNE      25$ ; LOOP IF NOT ZERO
11192 043316 032700 030000                BIT      #BIT12!BIT13,RO ; MEM MANAGE IN USE?
11193 043322 001402                BEQ      50$ ; NO - SKIP
11194 043324 005337 177572                DEC      @#SRO ; TURN IT OFF
11195 043330 50$:     MOV      (SP)+,R4 ; ;POP STACK INTO R4
11196 043330 012604                BR       16$
11197 043332 000427
11198
11199 ;
11200 ; ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
11201 043334 010537 001700 14$:     MOV      R5,DESHLD ; STORE DESTINATION
11202 043340 010437 001702                MOV      R4,SRCHLD ; SOURCE
11203 043344 014537 001202                MOV      -(R5),$REG10 ; LOAD GOOD WORD FOR REPORT
11204 043350 014437 001204                MOV      -(R4),$REG11 ; BAD WORD
11205 043354 010337 001206                MOV      R3,$REG12 ; WORD NUMBER
11206 043360 005301                DEC      R1 ; DEC COUNT LEFTFOR CONTINUATION
11207 043362 005203                INC      R3 ; BUMP BAD WORD NUMBER

```


M16

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35MACY11 30A(1052) 10-JAN-78 09:57 PAGE 207
DATA GENERATION AND COMPARE ROUTINE

SEQ 0207

```

11208 043364 010137 001706      MOV      R1,WRDCNT      ;STORE COUNT LEFT
11209 043370 010337 001704      MOV      R3,WRDNUM      ;      WORD NUM IN ERROR
11210 043374 032700 030000      BIT      #BIT12!BIT13,RO ;MEM MANAGE IS USE?
11211 043400 001402                BEQ      15$            ;NO - SKIP
11212 043402 005337 177572                DEC      @#SRO          ;TURN IT OFF
11213
11214 043406                15$:      MOV      (SP)+,R4      ;;POP STACK IN.O R4
11215 043406 012604                TST      (R4)+        ;ERROR RETURN
11216 043410 005724
11217
11218 043412                16$:      MOV      (SP)+,R5      ;;POP STACK INTO R5
11219 043412 012605                MOV      (SP)+,R3      ;;POP STACK INTO R3
11220 043414 012603                MOV      (SP)+,R1      ;;POP STACK INTO R1
11221 043416 012601                MOV      (SP)+,R0      ;;POP STACK INTO R0
11222 043420 012600                RTS      R4
11223 043422 000204
11224
11225 ;      DATA PATTERN PROCESSING ROUTINE
11226
11227 043424 032700 040000      17$:      BIT      #BIT14,RO      ;CONTINUE WITH COMPARE?
11228 043430 001402                BEQ      29$            ;NO - SKIP
11229 043432 010446                MOV      R4,-(SP)      ;STORE RETURN
11230 043434 000704                BR       28$            ;GO CONTINUE COMPARE
11231
11232 043436 012705 072414      29$:      MOV      #OBUFF,R5      ;GET DESTINATION
11233 043442 012703 070414      MOV      #IBUFF,R3      ;GET SOURCE
11234 043446 032700 030000      BIT      #BIT12!BIT13,RO ;USE MEM MANAGE?
11235 043452 001412                BEQ      21$            ;NO - SKIP
11236
11237 043454 012437 172354      MOV      (R4)+,@#KIPAR6 ;LOAD PAR FOR RELOCATION
11238 043460 032700 010000      BIT      #BIT12,RO      ;RELOCATE SOURCE?
11239 043464 001403                BEQ      20$            ;NO - SKIP
11240 043466 012705 140070      MOV      #140070,R5      ;SET DESTINATION TO USE PAR6 + OFFSET
11241 043472 000402                BR       21$            ;SKIP
11242 043474 012703 140070      20$:      MOV      #140070,R3      ;SET SOURCE TO USE PAR6 + OFFSET
11243
11244 043500 012401      21$:      MOV      (R4)+,R1      ;STORE COUNT
11245 043502 010446      MOV      R4,-(SP)      ;STORE RETURN
11246 043504 010304      MOV      R3,R4          ;PUT IN Ibuff POINTER
11247 043506 005003      CLR      R3            ;CLEAR R3 FOR WORD NUMBER COUNTER
11248 043510 032700 000077      BIT      #77,RO        ;ANY DATA PATTERN SPECIFIED?
11249 043514 001666                BEQ      13$            ;NO - GO DO COMPARE
11250
11251 ;      START OF GENERATION ROUTINE
11252
11253 043516 010537 001700      MOV      R5,DESHLD      ;STORE PARAMETERS FOR COMPARE
11254 043522 010437 001702      MOV      R4,SRCHLD
11255 043526 010337 001704      MOV      R3,WRDNUM
11256 043532 010137 001706      MOV      R1,WRDCNT
11257
11258 ;      CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
11259 ;      MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
11260 ;      WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
11261
11262 043536 032700 030000      BIT      #BIT12!BIT13,RO ;MEMORY MANAGEMENT REQUIRED?
11263 043542 001402                BEQ      33$            ;NO - SKIP

```

B01

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 208
DATA GENERATION AND COMPARE ROUTINE

SEQ 0208

11264	043544	005237	177572		INC	#SRO	:TURN IT ON
11265	043550	032700	002000	33\$:	BIT	#BIT10,R0	:GENERATE PATTERN IN Ibuff?
11266	043554	001403			BEG	32\$:NO - SKIP
11267	043556	010446			MOV	R4,-(SP)	:ELSE SWAP R4 AND R5
11268	043560	010504			MOV	R5,R4	
11269	043562	012605			MOV	(SP)+,R5	
11270							
11271	043564	122700	000001	32\$:	CMPB	#1,R0	:PATTERN 1 (ALL ZEROS)?
11272	043570	001004			BNE	55\$:NO - SKIP
11273	043572	005025		30\$:	CLR	(R5)+	:CLEAR WORD IN BUFF
11274	043574	005301			DEC	R1	:DEC WORD COUNT
11275	043576	001375			BNE	30\$:LOOP UNTIL WORD COUNT ZERO
11276	043600	000550			BR	22\$:EXIT BUILD
11277							
11278	043602	122700	000007	55\$:	CMPB	#7,R0	:PATTERN 7 (ALL ONES)?
11279	043606	001005			BNE	56\$:NO - SKIP
11280	043610	012725	177777	31\$:	MOV	#-1,(R5)+	:LOAD WORD IN BUFF
11281	043614	005301			DEC	R1	:DEC WORD COUNT
11282	043616	001374			BNE	31\$:LOOP UNTIL WORD COUNT ZERO
11283	043620	000540			BR	22\$:EXIT BUILD
11284							
11285	043622	122700	000002	56\$:	CMPB	#2,R0	:PATTERN 2 SET UP
11286	043626	001003			BNE	57\$	
11287	043630	012703	047522		MOV	#PAT02,R3	
11288	043634	000504			BR	70\$	
11289							
11290	043636	122700	000003	57\$:	CMPB	#3,R0	:PATTERN 3 SET UP
11291	043642	001003			BNE	58\$	
11292	043644	012703	047562		MOV	#PAT03,R3	
11293	043650	000476			BR	70\$	
11294							
11295	043652	122700	000004	58\$:	CMPB	#4,R0	:PATTERN 4 SET UP
11296	043656	001003			BNE	59\$	
11297	043660	012703	047622		MOV	#PAT04,R3	
11298	043664	000470			BR	70\$	
11299							
11300	043666	122700	000005	59\$:	CMPB	#5,R0	:PATTERN 5 SET UP
11301	043672	001003			BNE	60\$	
11302	043674	012703	047662		MOV	#PAT05,R3	
11303	043700	000462			BR	70\$	
11304							
11305	043702	122700	000006	60\$:	CMPB	#6,R0	:PATTERN 6 SET UP
11306	043706	001003			BNE	61\$	
11307	043710	012703	047722		MOV	#PAT06,R3	
11308	043714	000454			BR	70\$	
11309							
11310	043716	122700	000010	61\$:	CMPB	#10,R0	:PATTERN 10 SET UP
11311	043722	001003			BNE	62\$	
11312	043724	012703	047762		MOV	#PAT10,R3	
11313	043730	000446			BR	70\$	
11314							
11315	043732	122700	000011	62\$:	CMPB	#11,R0	:PATTERN 11 SET UP
11316	043736	001003			BNE	63\$	
11317	043740	012703	050022		MOV	#PAT11,R3	
11318	043744	000440			BR	70\$	
11319							

C01

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 209
DATA GENERATION AND COMPARE ROUTINE

SEQ 0209

11320	043746	122700	000012	63\$:	CMPB	#12,R0	;PATTERN 12 SET UP
11321	043752	001003			BNE	64\$	
11322	043754	012703	050062		MOV	#PAT12,R3	
11323	043760	000432			BR	70\$	
11324							
11325	043762	122700	000013	64\$:	CMPB	#13,R0	;PATTERN 13 SET UP
11326	043766	001003			BNE	65\$	
11327	043770	012703	050122		MOV	#PAT13,R3	
11328	043774	000424			BR	70\$	
11329							
11330	043776	122700	000014	65\$:	CMPB	#14,R0	;PATTERN 14 SET UP
11331	044002	001003			BNE	66\$	
11332	044004	012703	050162		MOV	#PAT14,R3	
11333	044010	000416			BR	70\$	
11334							
11335	044012	122700	000015	66\$:	CMPB	#15,R0	;PATTERN 15 SET UP
11336	044016	001003			BNE	67\$	
11337	044020	012703	050222		MOV	#PAT15,R3	
11338	044024	000410			BR	70\$	
11339							
11340	044026	122700	000016	67\$:	CMPB	#16,R0	;PATTERN 16 SET UP
11341	044032	001003			BNE	68\$	
11342	044034	012703	050262		MOV	#PAT16,R3	
11343	044040	000402			BR	70\$	
11344							
11345	044042	012703	050262	68\$:	MOV	#PAT16,R3	;SET UP FOR 16
11346							
11347	044046	032700	004000	70\$:	BIT	#BIT11,R0	;FIRST WORD REPEAT?
11348	044052	001020			BNE	73\$;YES - SKIP
11349	044054	010446			MOV	R4,-(SP)	;STORE R4
11350	044056	010046			MOV	R0,-(SP)	;STORE R0
11351	044060	012700	000020		MOV	#16,R0	;PRESET COUNT FOR PATTERN LENGTH
11352	044064	010504			MOV	R5,R4	;STORE START OF BUFF
11353							
11354	044066	012325		71\$:	MOV	(R3)+,(R5)+	;MOV WORD TO BUFF
11355	044070	005301			DEC	R1	;DEC WORD COUNT
11356	044072	001405			BEQ	74\$;EXIT IF ZERO
11357	044074	005300			DEC	R0	;DEC PAT LENGTH COUNT
11358	044076	001373			BNE	71\$;LOOP IF NOT ZERO
11359							
11360	044100	012425		72\$:	MOV	(R4)+,(R5)+	;REPEAT PATTERN IN BUFFER
11361	044102	005301			DEC	R1	;DEC WORD COUNT
11362	044104	001375			BNE	72\$;LOOP UNTIL WORD COUNT ZERO
11363							
11364	044106	012600		74\$:	MOV	(SP)+,R0	;RESTORE R0
11365	044110	012604			MOV	(SP)+,R4	;RESTORE R4
11366	044112	000403			BR	22\$;EXIT BUILD
11367							
11368	044114	011325		73\$:	MOV	(R3),(R5)+	;MOV THE SAME WORD INTO BUFFER
11369	044116	005301			DEC	R1	;DEC WORD COUNT
11370	044120	001375			BNE	73\$;LOOP UNTIL ZERO
11371							
11372	044122	032700	030000	22\$:	BIT	#BIT12!BIT13,R0	;MEMORY MANAGEMENT REQUIRED?
11373	044126	001402			BEQ	34\$;NO - SKIP
11374	044130	005337	177572		DEC	#SRO	;TURN OFF MEM MANAGEMENT
11375	044134	005700		34\$:	TST	R0	;IS COMPARE REQUIRED?

```

11376 044136 100012          BPL      23$          ;NO - SKIP
11377 044140 013705 001700  MOV      DESHLD,R5   ;RESTORE COMPARE PARAMETERS
11378 044144 013704 001702  MOV      SRCHLD,R4
11379 044150 013703 001704  MOV      WRDNUM,R3
11380 044154 013701 001706  MOV      WRDCNT,R1
11381 044160 000137 043272  JMP      13$          ;GO START COMPARE
11382 044164
11383 044164 012604          23$:  MOV      (SP)+,R4   ;:POP STACK INTO R4
11384 044166 000137 043412  JMP      16$          ;:GO TO EXIT
11385
11386 ;*****
11387 ;SBTTL PHASE LOCK LOOP CLOCK ADJUSTMENT ROUTINE
11388 ;* THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
11389 ;* PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
11390 ;* PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
11391 ;* THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
11392 ;* RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
11393 ;* SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
11394 ;*
11395 ;* THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
11396 ;*
11397 044172 104401 053314  ADJCLK: TYPE      ,OPR019          ;TYPE ADJUSTMENT INSTRUCTIONS
11398
11399 044176 012762 000040 000010  MOV      #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
11400 044204 013762 001626 000010  MOV      DRVNUM,RKCS2(R2) ;SELECT DRIVE
11401 044212 012762 000001 000000  MOV      #1,RKCS1(R2)
11402 044220 032762 000200 000000  5$:  BIT      #RDY,RKCS1(R2) ;WAIT FOR READY
11403 044226 001774          BEQ      5$
11404 044230 032737 100000 001664  BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
11405 044236 001402          BEQ      1$          ;NO - SKIP
11406 044240 005077 135444          CLR      #KWLADD        ;CLEAR INTERRUPT ENABLE
11407
11408 044244 012762 000040 000026  1$:  MOV      #DMD,RKMR1(R2) ;SET DIAG MODE
11409 044252 012701 000014          MOV      #12.,R1        ;SET A COUNT
11410 044256 005301          DEC      R1            ;DEC COUNT
11411 044260 001376          BNE     2$            ;LOOP UNTIL ZERO
11412 044262 012762 000000 000026  MOV      #0,RKMR1(R2)   ;CLEAR MR1
11413 044270 012701 000014          MOV      #12.,R1        ;SET COUNT
11414 044274 005301          DEC      R1            ;DEC COUNT
11415 044276 001376          BNE     3$            ;LOOP UNTIL ZERO
11416 044300 000761          BR       1$            ;RESTART LOOP
11417
11418 ;SBTTL CONTROLLED HALT SUBROUTINE
11419 ;* THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
11420 ;* SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
11421 ;* NECESSARY, CERTAIN CYLINDERS ARE REFORMATED. THE REFORMATTING
11422 ;* IS CONTROLLED BY THE LOCATION REFM T WHICH CONTAINS THE ADDRESS
11423 ;* OF THE CYLINDER TO BE REFORMATTED.
11424
11425 044302 012737 000112 001302  CTRHLT: MOV      #STN,$STEN          ;SET UP FOR HALT FAIL
11426
11427 044310 104416          TSSINIT          ;CLEAR SUBSYSTEM
11428 044312 104003          ERROR      3          ;BAD INIT ERROR
11429
11430 044314 113700 001102          MOV      $STNM,RO      ;GET CURRENT TEST NUMBER
11431 044320 042700 177400          BIC      #177400,RO    ;CLEAR UNUSED BITS

```

11432	044324	022700	000003		CMP	#3,RO		;TEST IF TEST NUMBER 3
11433	044330	001464			BEQ	PROGEND		;GO TO HALT PROG
11434	044332	004437	036252		JSR	R4,LRLoad		;LOAD "L" REGS
11435	044336	000113			RECAL			;RECAL
11436	044340	000000			0			;0 WORDS
11437	044342	000000			0			;0 IS BUFF ADDRESS
11438	044344	000			.BYTE	0		;SECTOR 0
11439	044345	000			.BYTE	0		;TRACK 0
11440	044346	000000			0			;CYLINDER 0
11441								
11442	044350	104417			TLOADRK			;LOAD RK REGS
11443	044352	104423			TWAT16			;WAIT FOR INTERRUPT
11444	044354	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
11445								
11446	044356	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11447	044360	104004			ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11448								
11449	044362	005037	001662		CLR	INTSET		;CLEAR INTERRUPT FLAG
11450	044366	104437			TWAT8S			;WAIT FOR SECOND INTERRUPT
11451	044370	104002			ERROR	2		
11452								
11453	044372	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11454	044374	104004			ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11455								
11456	044376	104416			TSSINIT			;CLEAR SUBSYSTEM
11457	044400	104003			ERROR	3		;BAD INIT ERROR
11458								
11459	044402	005737	001676		TST	REFMT		;TEST IF REFORMAT REQUIRED
11460	044406	001435			BEQ	PROGEND		;NO - GO TO HALT
11461	044410	104401	053457		TYPE	,OPRO20		;TYPE MESSAGE
11462								
11463	044414	004437	036252		JSR	R4,LRLoad		;LOAD "L" REGS
11464	044420	000127			WRHEAD			;WRHEAD
11465	044422	177676			-102			; -102 WORDS
11466	044424	072414			OBUFF			;OBUFF IS BUFF ADDRESS
11467	044426	000			.BYTE	0		;SECTOR 0
11468	044427	000			.BYTE	0		;TRACK 0
11469	044430	000312			312			;CYLINDER 312
11470								
11471	044432	005737	001676		TST	REFMT		;TEST IF CYL 0
11472	044436	100002			BPL	SS		;NO - SKIP
11473	044440	005037	001614		CLR	L.DCYL		;ELSE LOAD FOR CYL 0
11474	044444	004437	042636	SS:	JSR	R4,GENCOM		;GENERATE HEADERS
11475	044450	001200			1200			
11476								
11477	044452	104417			TLOADRK			;LOAD RK REGS
11478	044454	104434			TWAT159			;WAIT FOR INTERRUPT
11479	044456	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
11480								
11481	044460	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11482	044462	104004			ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11483								
11484	044464	122737	000002 001607		CMPB	#2,L.DT		;TEST IF TRACK 2 FORMATTED
11485	044472	001403			BEQ	PROGEND		;YES - SKIP
11486	044474	105237	001607		INCB	L.DT		;ELSE BUMP TRACK
11487	044500	000761			BR	SS		;DO NEXT TRACK

```

11488
11489 044502 104401 053553
11490 044506 012706 001100
11491 044512 105037 001103
11492 044516 005037 001264
11493 044522 005737 000042
11494 044526 001404
11495 044530 005037 032450
11496 044534 000137 032422
11497
11498 044540 000000
11499 044542 000137 001776
11500
11501
11502
11503
11504 044546 000240
11505 044550 104401 053614
11506 044554 000137 044502
11507
11508
11509
11510
11511
11512
11513
11514
11515
11516
11517
11518
11519
11520
11521
11522
11523
11524 044560 105737 001157
11525 044564 100002
11526 044566 000000
11527 044570 000430
11528 044572 010046
11529 044574 017600 000002
11530 044600 122737 000001 001316
11531 044606 001011
11532 044610 132737 000100 001317
11533 044616 001405
11534 044620 010037 044630
11535 044624 004737 033752
11536 044630 000000
11537 044632 132737 000040 001317
11538 044640 001003
11539 044642 112046
11540 044644 001005
11541 044646 005726
11542 044650 012600
11543 044652 062716 000002

```

```

PROGEND:      TYPE      OPR021 ;TYPE HALT MESSAGE
              MOV      #STACK,SP ;CLEAR STACK
              CLR      $ERFLG    ;CLEAR ERROR FLAG
              CLR      $ESCAPE   ;CLEAR ESCAPE
              TST      @#42      ;TEST IF MONITOR PRESENT
              BEQ      10$       ;NO - SKIP
              CLR      $EOPCT    ;SET FOR END OF PROGRAM
              JMP      $EOP      ;GO TO END OF PASS

10$:          HALT          ;HALT PROGRAM
              JMP      START1    ;GO TO RESTART IF CONTINUE

.SBTTL  HALT FAIL ROUTINE
;*      THIS ROUTINE IS ENTERED IF A HARDWARE ERROR IS DETECTED WHEN
;*      THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
ABTFAIL:     NOP
              TYPE      OPR022   ;TYPE HALT FAIL MESSAGE
              JMP      PROGEND    ;GO STOP PROGRAM

.SBTTL  TYPE ROUTINE
;*****
;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;*
;*CALL:
;*1) USING A TRAP INSTRUCTION
;*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;*OR
;*      TYPE      MESADR
;*
$TYPE:       TSTB      $TPFLG    ;; IS THERE A TERMINAL?
              BPL      1$        ;; BR IF YES
              HALT          ;; HALT HERE IF NO TERMINAL
              BR      3$         ;; LEAVE
1$:          MOV      RO, -(SP)   ;; SAVE RO
              MOV      @2(SP),RO  ;; GET ADDRESS OF ASCIZ STRING
              CMPB     #APTENV,$ENV ;; RUNNING IN APT MODE
              BNE     62$        ;; NO, GO CHECK FOR APT CONSOLE
              BITB     #APTSPool,$ENVM ;; SPOOL MESSAGE TO APT
              BEQ     62$        ;; NO, GO CHECK FOR CONSOLE
              MOV      RO,61$     ;; SETUP MESSAGE ADDRESS FOR APT
              JSR     PC,$ATY3    ;; SPOOL MESSAGE TO APT
              .WORD   0          ;; MESSAGE ADDRESS
              BITB     #APTCsup,$ENVM ;; APT CONSOLE SUPPRESSED
              BNE     60$        ;; YES, SKIP TYPE OUT
              MOVB     (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
              BNE     4$         ;; BR IF IT ISN'T THE TERMINATOR
              TST     (SP)+      ;; IF TERMINATOR POP IT OFF THE STACK
              MOV      (SP)+,RO   ;; RESTORE RO
              ADD     #2,(SP)    ;; ADJUST RETURN PC
61$:        .WORD   0
62$:        BITB     #APTCsup,$ENVM
60$:        BNE     60$
2$:         MOVB     (RO)+, -(SP)
4$:         BNE     4$
60$:        MOV      (SP)+,RO
3$:         ADD     #2,(SP)

```

```

11544 044656 000002          RTI          ;;RETURN
11545 044660 122716 000011 4$: CMPB    #HT,(SP)      ;;BRANCH IF <HT>
11546 044664 001430          BEQ      8$              ;;
11547 044666 122716 000200  CMPB    #CRLF,(SP)     ;;BRANCH IF NOT <CRLF>
11548 044672 001006          BNE      5$              ;;
11549 044674 005726          TST     (SP)+          ;;POP <CR><LF> EQUIV
11550 044676 104401          TYPE                    ;;TYPE A CR AND LF
11551 044700 001273          $CRLF
11552 044702 105037 045036  CLRB    $CHARCNT       ;;CLEAR CHARACTER COUNT
11553 044706 000755          BR      2$              ;;GET NEXT CHARACTER
11554 044710 004737 044772 5$: JSR     PC,$TYPEPC    ;;GO TYPE THIS CHARACTER
11555 044714 123726 001156 6$: CMPB    $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
11556 044720 001350          BNE      2$              ;;IF NO GO GET NEXT CHAR.
11557 044722 013746 001154  MOV     $NULL,-(SP)    ;;GET # OF FILLER CHARS. NEEDED
11558                                     AND THE NULL CHAR.
11559 044726 105366 000001 7$: DECB    1(SP)        ;;DOES A NULL NEED TO BE TYPED?
11560 044732 002770          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
11561 044734 004737 044772  JSR     PC,$TYPEPC    ;;GO TYPE A NULL
11562 044740 105337 045036  DECB    $CHARCNT       ;;DO NOT COUNT AS A COUNT
11563 044744 000770          BR      7$              ;;LOOP
11564
11565 ;HORIZONTAL TAB PROCESSOR
11566
11567 044746 112716 000040 8$: MOVB    #' ,(SP)    ;;REPLACE TAB WITH SPACE
11568 044752 004737 044772 9$: JSR     PC,$TYPEPC    ;;TYPE A SPACE
11569 044756 132737 000007 045036 BITB    #',$CHARCNT    ;;BRANCH IF NOT AT
11570 044764 001372          BNE      9$              ;;TAB STOP
11571 044766 005726          TST     (SP)+          ;;POP SPACE OFF STACK
11572 044770 000724          BR      2$              ;;GET NEXT CHARACTER
11573 044772 105777 134152 $TYPEPC: TSTB   $STPS     ;;WAIT UNTIL PRINTER IS READY
11574 044776 100375          BPL     $TYPEPC
11575 045000 116677 000002 134144 MOVB    2(SP),$STPB    ;;LOAD CHAR TO BE TYPED INTO DATA REG.
11576 045006 122766 000015 000002 CMPB    #CR,2(SP)     ;;IS CHARACTER A CARRIAGE RETURN?
11577 045014 001003          BNE      1$              ;;BRANCH IF NO
11578 045016 105037 045036  CLRB    $CHARCNT       ;;YES--CLEAR CHARACTER COUNT
11579 045022 000406          BR      $TYPEPC        ;;EXIT
11580 045024 122766 000012 000002 1$: CMPB    #LF,2(SP)    ;;IS CHARACTER A LINE FEED?
11581 045032 001402          BEQ     $TYPEPC        ;;BRANCH IF YES
11582 045034 105227          INCB   (PC)+          ;;COUNT THE CHARACTER
11583 045036 000000  $CHARCNT: .WORD 0     ;;CHARACTER COUNT STORAGE
11584 045040 000207  $TYPEPC: RTS     PC
11585
11586 .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11587
11588 ;*****
11589 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
11590 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11591 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11592 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11593 ;*REPLACED WITH SPACES.
11594 ;*CALL:
11595 ;*      MOV     NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
11596 ;*      TYPDS                    ;;GO TO THE ROUTINE
11597
11598 045042 $TYPDS: MOV     RO,-(SP)      ;;PUSH RO ON STACK
11599 045042 010046

```

```

11600 045044 010146      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
11601 045046 010246      MOV      R2,-(SP)      ;; PUSH R2 ON STACK
11602 045050 010346      MOV      R3,-(SP)      ;; PUSH R3 ON STACK
11603 045052 010546      MOV      R5,-(SP)      ;; PUSH R5 ON STACK
11604 045054 012746      MOV      #20200,-(SP)  ;; SET BLANK SWITCH AND SIGN
11605 045060 016605      MOV      20(SP),R5     ;; GET THE INPUT NUMBER
11606 045064 100004      BPL      1$           ;; BR IF INPUT IS POS.
11607 045066 005405      NEG      R5           ;; MAKE THE BINARY NUMBER POS.
11608 045070 112766      MOV      #'-,1(SP)    ;; MAKE THE ASCII NUMBER NEG.
11609 045076 005000      CLR      R0           ;; ZERO THE CONSTANTS INDEX
11610 045100 012703      MOV      #SDBLK,R3    ;; SETUP THE OUTPUT POINTER
11611 045104 112723      MOV      #' ,(R3)+    ;; SET THE FIRST CHARACTER TO A BLANK
11612 045110 005002      CLR      R2           ;; CLEAR THE BCD NUMBER
11613 045112 016001      MOV      SDBTL(R0),R1 ;; GET THE CONSTANT
11614 045116 160105      SUB      R1,R5        ;; FORM THIS BCD DIGIT
11615 045120 002402      BLT      4$           ;; BR IF DONE
11616 045122 005202      INC      R2           ;; INCREASE THE BCD DIGIT BY 1
11617 045124 000774      BR       3$
11618 045126 060105      ADD      R1,R5        ;; ADD BACK THE CONSTANT
11619 045130 005702      TST      R2           ;; CHECK IF BCD DIGIT=0
11620 045132 001002      BNE      5$           ;; FALL THROUGH IF 0
11621 045134 105716      TSTB    (SP)         ;; STILL DOING LEADING 0'S?
11622 045136 100407      BMI      7$           ;; BR IF YES
11623 045140 106316      ASLB    (SP)         ;; MSD?
11624 045142 103003      BCC      6$           ;; BR IF NO
11625 045144 116663      MOV      1(SP),-1(R3) ;; YES--SET THE SIGN
11626 045152 052702      BIS      #'0,R2       ;; MAKE THE BCD DIGIT ASCII
11627 045156 052702      BIS      #' ,R2       ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
11628 045162 110223      MOV      R2,(R3)+    ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
11629 045164 005720      TST      (R0)+       ;; JUST INCREMENTING
11630 045166 020027      CMP      R0,#10      ;; CHECK THE TABLE INDEX
11631 045172 002746      BLT      2$           ;; GO DO THE NEXT DIGIT
11632 045174 003002      BGT      8$           ;; GO TO EXIT
11633 045176 010502      MOV      R5,R2       ;; GET THE LSD
11634 045200 000764      BR       6$           ;; GO CHANGE TO ASCII
11635 045202 105726      TSTB    (SP)+        ;; WAS THE LSD THE FIRST NON-ZERO?
11636 045204 100003      BPL      9$           ;; BR IF NO
11637 045206 116663      MOV      -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
11638 045214 105013      CLRB    (R3)         ;; SET THE TERMINATOR
11639 045216 012605      MOV      (SP)+,R5     ;; POP STACK INTO R5
11640 045220 012603      MOV      (SP)+,R3     ;; POP STACK INTO R3
11641 045222 012602      MOV      (SP)+,R2     ;; POP STACK INTO R2
11642 045224 012601      MOV      (SP)+,R1     ;; POP STACK INTO R1
11643 045226 012600      MOV      (SP)+,R0     ;; POP STACK INTO R0
11644 045230 104401      TYPE    #SDBLK       ;; NOW TYPE THE NUMBER
11645 045234 016666      MOV      2(SP),4(SP)  ;; ADJUST THE STACK
11646 045242 012616      MOV      (SP)+,(SP)
11647 045244 000002      RTI
11648 045246 023420      SDBTL: 10000.
11649 045250 001750      1000.
11650 045252 000144      100.
11651 045254 000012      10.
11652 045256 000004      SDBLK: .BLKW 4
11653      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
11654
11655      ;; *****

```


IO1

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 215
BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0215

```

11656      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11657      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
11658      ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11659      ;*CALL:
11660      ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
11661      ;*      TYPOS          ;;CALL FOR TYPEOUT
11662      ;*      .BYTE  N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
11663      ;*      .BYTE  M          ;;M=1 OR 0
11664      ;*                                  ;;1=TYPE LEADING ZEROS
11665      ;*                                  ;;0=SUPPRESS LEADING ZEROS
11666      ;*
11667      ;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11668      ;*$TYPOS OR $TYPOC
11669      ;*CALL:
11670      ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
11671      ;*      TYPON          ;;CALL FOR TYPEOUT
11672      ;*
11673      ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11674      ;*CALL:
11675      ;*      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
11676      ;*      TYPOC          ;;CALL FOR TYPEOUT
11677      ;*
11678      045266 017646 000000 045511 $TYPOS: MOV      2(SP),-(SP)          ;;PICKUP THE MODE
11679      045272 116637 000001          MOVVB   1(SP),$OFILL          ;;LOAD ZERO FILL SWITCH
11680      045300 112637 045513          MOVVB   (SP)+,$OMODE+1      ;;NUMBER OF DIGITS TO TYPE
11681      045304 062716 000002          ADD     #2,(SP)          ;;ADJUST RETURN ADDRESS
11682      045310 000406          BR      $TYPON
11683      045312 112737 000001 045511 $TYPOC: MOVVB   #1,$OFILL          ;;SET THE ZERO FILL SWITCH
11684      045320 112737 000006 045513          MOVVB   #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
11685      045326 112737 000005 045510 $TYPON: MOVVB   #5,$OCNT          ;;SET THE ITERATION COUNT
11686      045334 010346          MOV     R3,-(SP)          ;;SAVE R3
11687      045336 010446          MOV     R4,-(SP)          ;;SAVE R4
11688      045340 010546          MOV     R5,-(SP)          ;;SAVE R5
11689      045342 113704 045513          MOVVB   $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
11690      045346 005404          NEG     R4
11691      045350 062704 000006          ADD     #6,R4          ;;SUBTRACT IT FOR MAX. ALLOWED
11692      045354 110437 045512          MOVVB   R4,$OMODE          ;;SAVE IT FOR USE
11693      045360 113704 045511          MOVVB   $OFILL,R4          ;;GET THE ZERO FILL SWITCH
11694      045364 016605 000012          MOV     12(SP),R5        ;;PICKUP THE INPUT NUMBER
11695      045370 005003          CLR     R3          ;;CLEAR THE OUTPUT WORD
11696      045372 006105          1$:    ROL     R5          ;;ROTATE MSB INTO "C"
11697      045374 070404          BR      3$          ;;GO DO MSB
11698      045376 006105          2$:    ROL     R5          ;;FORM THIS DIGIT
11699      045400 006105          ROL     R5
11700      045402 006105          ROL     R5
11701      045404 010503          MOV     R5,R3
11702      045406 006103          3$:    ROL     R3          ;;GET LSB OF THIS DIGIT
11703      045410 105337 045512          DECB   $OMODE          ;;TYPE THIS DIGIT?
11704      045414 100016          BPL    7$          ;;BR IF NO
11705      045416 042703 177770          BIC    #177770,R3      ;;GET RID OF JUNK
11706      045422 001002          BNE    4$          ;;TEST FOR 0
11707      045424 005704          TST    R4          ;;SUPPRESS THIS 0?
11708      045426 001403          BEQ    5$          ;;BR IF YES
11709      045430 005204          4$:    INC     R4          ;;DON'T SUPPRESS ANYMORE 0'S
11710      045432 052703 000060          BIS    #'0,R3          ;;MAKE THIS DIGIT ASCII
11711      045436 052703 000040          5$:    BIS    #' ,R3      ;;MAKE ASCII IF NOT ALREADY

```

```

11712 045442 110337 045506          MOVB    R3,B$          ;; SAVE FOR TYPING
11713 045446 104401 045506          TYPE    B$           ;; GO TYPE THIS DIGIT
11714 045452 105337 045510          7$:    DECB    $OCNT  ;; COUNT BY 1
11715 045456 003347                BGT     2$           ;; BR IF MORE TO DO
11716 045460 002402                BLT     6$           ;; BR IF DONE
11717 045462 005204                INC     R4           ;; INSURE LAST DIGIT ISN'T A BLANK
11718 045464 000744                BR      2$           ;; GO DO THE LAST DIGIT
11719 045466 012605          6$:    MOV     (SP)+,R5  ;; RESTORE R5
11720 045470 012604                MOV     (SP)+,R4    ;; RESTORE R4
11721 045472 012603                MOV     (SP)+,R3    ;; RESTORE R3
11722 045474 016666 000002 000004  MOV     2(SP),4(SP)  ;; SET THE STACK FOR RETURNING
11723 045502 012616                MOV     (SP)+,(SP)
11724 045504 000002                RTI                    ;; RETURN
11725 045506          000          8$:    .BYTE    0          ;; STORAGE FOR ASCII DIGIT
11726 045507          000                .BYTE    0          ;; TERMINATOR FOR TYPE ROUTINE
11727 045510          000          $OCNT: .BYTE    0          ;; OCTAL DIGIT COUNTER
11728 045511          000          $OFILL: .BYTE    0        ;; ZERO FILL SWITCH
11729 045512 000000          $OMODE: .WORD    0        ;; NUMBER OF DIGITS TO TYPE
11730
11731          .SBTTL  TTY INPUT ROUTINE
11732
11733          ;;*****
11734 045514 000000          .ENABL  LSB          ;;
11735 045516 000000          $TKCNT: .WORD    0          ;; NUMBER OF ITEMS IN QUEUE
11736 045520 000000          $TKQIN: .WORD    0          ;; INPUT POINTER
11737 045522 000001          $TKQOUT: .WORD   0          ;; OUTPUT POINTER
11738          045523          $TKQSR: .BLKB   1          ;; TTY KEYBOARD QUEUE
11739          045524          $TKQEND=.
11740          .EVEN
11741
11742          ;*TK INITIALIZE ROUTINE
11743          ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11744          ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11745
11746          ;*CALL:
11747          ;*      JSR      PC,$TKINT
11748          ;*      RETURN
11749 045524 005037 045514          $TKINT: CLR     $TKCNT      ;; CLEAR COUNT OF ITEMS IN QUEUE
11750 045530 012737 045522 045516  MOV     # $TKQSR,$TKQIN  ;; MOVE THE STARTING ADDRESS OF THE
11751 045536 013737 045516 045520  MOV     $TKQIN,$TKQOUT  ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
11752 045544 012737 045574 000060  MOV     # $TKSRV,$TKVEC ;; INITIALIZE THE KEYBOARD VECTOR
11753 045552 012737 000200 000062  MOV     #200,$TKVEC+2  ;; "BR" LEVEL 4
11754 045560 005777 133362          TST     $TKB           ;; CLEAR DONE FLAG
11755 045564 012777 000100 133352  MOV     #100,$TKS      ;; ENABLE TTY KEYBOARD INTERRUPT
11756 045572 000207                RTS     PC            ;; RETURN TO CALLER
11757
11758          ;*TK SERVICE ROUTINE
11759          ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
11760          ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
11761          ;*IT IN THE QUEUE.
11762          ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
11763          ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
11764
11765 045574 117746 133346          $TKSRV: MOVB    @ $TKB,-(SP)  ;; PICKUP THE CHARACTER
11766 045600 042716 177600          BIC     #↑C177,(SP)      ;; STRIP THE JUNK
11767 045604 021627 000003          CMP     (SP),#3        ;; IS IT A CONTROL C?

```

```

11768 045610 001007          BNE 1$          ;; BRANCH IF NO
11769 045612 104401 046710   TYPE ,SCNTLC   ;; TYPE A CONTROL-C (↑C)
11770 045616 004737 045524   JSR PC,STKINT ;; INIT THE KEYBOARD
11771 045622 005726          TST (SP)+      ;; CLEAN UP STACK
11772 045624 000137 044302   JMP CTRHLT    ;; CONTROL C RESTART
11773 045630 021627 000007   1$: CMP (SP),#7  ;; IS IT A CONTROL G?
11774 045634 001004          BNE 2$          ;; BRANCH IF NO
11775 045636 022737 000176 001140 CMP #SWREG,SWR ;; IS SOFT-SWR SELECTED?
11776 045644 001500          BEQ 6$          ;; GO TO SWR CHANGE
11777
11778 045646          2$:
11779 045646 022737 000001 045514 CMP #1,STKCNT  ;; IS THE QUEUE FULL?
11780 045654 001004          BNE 3$          ;; BRANCH IF NO
11781 045656 104401 001266   TYPE ,SBELL   ;; RING THE TTY BELL
11782 045662 005726          TST (SP)+      ;; CLEAN CHARACTER OFF OF STACK
11783 045664 000451          BR 5$          ;; EXIT
11784 045666 021627 000023   3$: CMP (SP),#23  ;; IS IT A CONTROL-S?
11785 045672 001021          BNE 32$         ;; BRANCH IF NO
11786 045674 005077 133244   CLR @STKS    ;; DISABLE TTY KEYBOARD INTERRUPTS
11787 045700 005726          TST (SP)+      ;; CLEAN CHAR OFF STACK
11788 045702 105777 133236   31$: TSTB @STKS   ;; WAIT FOR A CHAR
11789 045706 100375          BPL 31$        ;; LOOP UNTIL ITS THERE
11790 045710 117746 133232   MOVB @STKB,-(SP) ;; GET THE CHARACTER
11791 045714 042716 177600   BIC #↑C177,(SP) ;; MAKE IT 7-BIT ASCII
11792 045720 022627 000021   CMP (SP)+,#21 ;; IS IT A CONTROL-Q?
11793 045724 001366          BNE 31$        ;; BRANCH IF NO
11794 045726 012777 000100 133210 MOV #100,@STKS ;; REENABLE TTY KEYBOARD INTERRUPTS
11795 045734 000002          RTI           ;; RETURN
11796 045736 005237 045514   32$: INC STKCNT   ;; COUNT THIS CHARACTER
11797 045742 021627 000140   CMP (SP),#140 ;; IS IT UPPER CASE?
11798 045746 002405          BLT 4$         ;; BRANCH IF YES
11799 045750 021627 000175   CMP (SP),#175 ;; IS IT A SPECIAL CHAR?
11800 045754 003002          BGT 4$         ;; BRANCH IF YES
11801 045756 042716 000040   BIC #40,(SP)  ;; MAKE IT UPPER CASE
11802 045762 112677 177530   4$: MOVB (SP)+,@STKQIN ;; AND PUT IT IN QUEUE
11803 045766 005237 045516   INC STKQIN    ;; UPDATE THE POINTER
11804 045772 023727 045516 045523 CMP STKQIN,#STKQEND ;; GO OFF THE END?
11805 046000 001003          BNE 5$         ;; BRANCH IF NO
11806 046002 012737 045522 045516 MOV #STKQSR,STKQIN ;; RESET THE POINTER
11807 046010 000002   5$: RTI           ;; RETURN
11808
11809
11810 ;; *****
11811 ;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11812 ;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
11813 ;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11814 ;; *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11814 046012 022737 000176 001140 $CKSWR: CMP #SWREG,SWR ;; IS THE SOFT-SWR SELECTED
11815 046020 001124          BNE 15$        ;; EXIT IF NOT
11816 046022 105777 133116   TSTB @STKS   ;; IS A CHAR WAITING?
11817 046026 100121          BPL 15$        ;; IF NOT, EXIT
11818 046030 117746 133112   MOVB @STKB,-(SP) ;; YES
11819 046034 042716 177600   BIC #↑C177,(SP) ;; MAKE IT 7-BIT ASCII
11820 046040 021627 000007   CMP (SP),#7  ;; IS IT A CONTROL-G?
11821 046044 001300          BNE 2$         ;; IF NOT, PUT IT IN THE TTY QUEUE
11822
11823

```

```

11824      ;:*****
11825      ;:CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11826      ;:ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11827      ;:CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11828 046046 123727 001134 000001 6$:      CMPB      $AUTOB,#1      ;:ARE WE RUNNING IN AUTO-MODE?
11829 046054 001674      BEQ        2$              ;:BRANCH IF YES
11830 046056 005726      TST        (SP)+          ;:CLEAR CONTROL-G OFF STACK
11831 046060 004737 045524      JSR        PC,$TKINT      ;:FLUSH THE TTY INPUT QUEUE
11832 046064 005077 133054      CLR        @STKS          ;:DISABLE TTY KEYBOARD INTERRUPTS
11833 046070 112737 000001 001135      MOVB       #1,$INTAG      ;:SET INTERRUPT MODE INDICATOR
11834
11835 046076 104401 046722      TYPE      ,SCNTLG        ;:ECHO THE CONTROL-G (↑G)
11836 046102 104401 046727      SGT$SWR: TYPE      $MSWR      ;:TYPE CURRENT CONTENTS
11837 046106 013746 000176      MOV        $WREG,-(SP)    ;:SAVE SWREG FOR TYPEOUT
11838 046112 104402      TYPEPC    ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
11839 046114 104401 046740      TYPE      ,SMNEW          ;:PROMPT FOR NEW SWR
11840 046120 005046      19$:      CLR        -(SP)        ;:CLEAR COUNTER
11841 046122 005046      CLR        -(SP)        ;:THE NEW SWR
11842 046124 105777 133014      7$:      TSTB       @STKS          ;:CHAR THERE?
11843 046130 100375      BPL        7$            ;:IF NOT TRY AGAIN
11844
11845 046132 117746 133010      MOVB       @STKB,-(SP)    ;:PICK UP CHAR
11846 046136 042716 177600      BIC        #1C177,(SP)   ;:MAKE IT 7-BIT ASCII
11847
11848 046142 021627 000003      CMP        (SP),#3        ;:IS IT A CONTROL-C?
11849 046146 001015      BNE        9$            ;:BRANCH IF NOT
11850 046150 104401 046710      TYPE      ,SCNTLC        ;:YES, ECHO CONTROL-C (↑C)
11851 046154 062706 000006      ADD        #6,SP          ;:CLEAN UP STACK
11852 046160 123727 001135 000001      CMPB       $INTAG,#1      ;:REENABLE TTY KEYBOARD INTERRUPTS?
11853 046166 001003      BNE        8$            ;:BRANCH IF NO
11854 046170 012777 000100 132746      MOV        #100,@STKS     ;:ALLOW TTY KEYBOARD INTERRUPTS
11855 046176 000137 044302      8$:      JMP         CTRHLT        ;:CONTROL-C RESTART
11856
11857
11858 046202 021627 000025      9$:      CMP        (SP),#25       ;:IS IT A CONTROL-U?
11859 046206 001005      BNE        10$           ;:BRANCH IF NOT
11860 046210 104401 046715      TYPE      ,SCNTLU        ;:YES, ECHO CONTROL-U (↑U)
11861 046214 062706 000006      20$:     ADD        #6,SP          ;:IGNORE PREVIOUS INPUT
11862 046220 000737      BR         19$           ;:LET'S TRY IT AGAIN
11863
11864
11865 046222 021627 000015      10$:     CMP        (SP),#15       ;:IS IT A <CR>?
11866 046226 001022      BNE        16$           ;:BRANCH IF NO
11867 046230 005766 000004      TST        4(SP)         ;:YES, IS IT THE FIRST CHAR?
11868 046234 001403      BEQ        11$           ;:BRANCH IF YES
11869 046236 016677 000002 132674      MOV        2(SP),@SWR     ;:SAVE NEW SWR
11870 046244 062706 000006      11$:     ADD        #6,SP          ;:CLEAR UP STACK
11871 046250 104401 001273      14$:     TYPE      ,SCRLF         ;:ECHO <CR> AND <LF>
11872 046254 123727 001135 000001      CMPB       $INTAG,#1      ;:RE-ENABLE TTY KBD INTERRUPTS?
11873 046262 001003      BNE        15$           ;:BRANCH IF NOT
11874 046264 012777 000100 132652      MOV        #100,@STKS     ;:RE-ENABLE TTY KBD INTERRUPTS
11875 046272 000002      15$:     RTI                    ;:RETURN
11876 046274 004737 044772      16$:     JSR        PC,$TYPEC      ;:ECHO CHAR
11877 046300 021627 000060      CMP        (SP),#60       ;:CHAR < 0?
11878 046304 002420      BLT        18$           ;:BRANCH IF YES
11879 046306 021627 000067      CMP        (SP),#67       ;:CHAR > 7?

```

11880 046312 003015
 11881 046314 042726 000060
 11882 046320 005766 000002
 11883 046324 001403
 11884 046326 006316
 11885 046330 006316
 11886 046332 006316
 11887 046334 005266 000002
 11888 046340 056616 177776
 11889 046344 000667
 11890 046346 104401 001272
 11891 046352 000720

BGT 18\$
 BIC #60,(SP)+
 TST 2(SP)
 BEQ 17\$
 ASL (SP)
 ASL (SP)
 ASL (SP)
 17\$: INC 2(SP)
 BIS -2(SP),(SP)
 BR 7\$
 18\$: TYPE \$QUES
 BR 20\$
 .DSABL LSB

:: BRANCH IF YES
 :: STRIP-OFF ASCII
 :: IS THIS THE FIRST CHAR
 :: BRANCH IF YES
 :: NO, SHIFT PRESENT
 :: CHAR OVER TO MAKE
 :: ROOM FOR NEW ONE.
 :: KEEP COUNT OF CHAR
 :: SET IN NEW CHAR
 :: GET THE NEXT ONE
 :: TYPE ?<CR><LF>
 :: SIMULATE CONTROL-U

11892
 11893
 11894
 11895
 11896
 11897
 11898
 11899
 11900
 11901
 11902

:: *****
 :: THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 :: *CALL:
 :: * RDCHR :: GET A CHARACTER FROM THE QUEUE
 :: * RETURN HERE :: CHARACTER IS ON THE STACK
 :: * :: WITH PARITY BIT STRIPPED OFF
 :: *

11903 046354 011646
 11904 046356 016666 000004 000002
 11905 046364 005066 000004
 11906 046370 005046
 11907 046372 012746 046400
 11908 046376 000002
 11909 046400
 11910 046400 005737 045514
 11911 046404 001775
 11912 046406 005337 045514
 11913 046412 117766 177102 000004
 11914 046420 005237 045520
 11915 046424 023727 045520 045523
 11916 046432 001003
 11917 046434 012737 045522 045520
 11918 046442 000002

\$RDCHR: MOV (SP),-(SP)
 MOV 4(SP),2(SP)
 CLR 4(SP)
 CLR -(SP)
 MOV #64\$,-(SP)
 RTI
 64\$:
 1\$: TST \$TKCNT
 BEQ 1\$
 DEC \$TKCNT
 MOVB 2\$TKQOUT,4(SP)
 INC \$TKQOUT
 CMP \$TKQOUT,#\$TKQEND
 BNE 2\$
 MOV #\$TKQSRRT,\$TKQOUT
 RTI
 2\$:

:: PUSH DOWN THE PC AND
 :: THE PS
 :: GET READY FOR A CHARACTER
 :: PUT NEW PS ON STACK
 :: PUT NEW PC ON STACK
 :: POP NEW PC AND PS
 :: WAIT ON A CHARACTER
 :: DECREMENT THE COUNTER
 :: GET ONE CHARACTER
 :: UPDATE THE POINTER
 :: DID IT GO OFF OF THE END?
 :: BRANCH IF NO
 :: RESET THE POINTER
 :: RETURN

11919
 11920
 11921
 11922
 11923
 11924
 11925

:: *****
 :: THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 :: *CALL:
 :: * RDLIN :: INPUT A STRING FROM THE TTY
 :: * RETURN HERE :: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 :: * :: TERMINATOR WILL BE A BYTE OF ALL 0'S
 :: *

11926 046444 010346
 11927 046446 005046
 11928 046450 012703 046700
 11929 046454 022703 046710
 11930 046460 101456
 11931 046462 104410
 11932 046464 112613
 11933 046466 122713 000177
 11934 046472 001022
 11935 046474 005716

\$RDLIN: MOV R3,-(SP)
 CLR -(SP)
 1\$: MOV \$TTYIN,R3
 2\$: CMP \$TTYIN+8.,R3
 BLOS 4\$
 RDCHR
 MOVB (SP)+,(R3)
 10\$: CMPB #177,(R3)
 BNE 5\$
 TST (SP)

:: SAVE R3
 :: CLEAR THE RUBOUT KEY
 :: GET ADDRESS
 :: BUFFER FULL?
 :: BR IF YES
 :: GO READ ONE CHARACTER FROM THE TTY
 :: GET CHARACTER
 :: IS IT A RUBOUT
 :: BR IF NO
 :: IS THIS THE FIRST RUBOUT?

```

11936 046476 001007          BNE      6$          ;;BR IF NO
11937 046500 112737 000134 046676  MOVB   #' \, 9$    ;;TYPE A BACK SLASH
11938 046506 104401 046676      TYPE   5$
11939 046512 012716 177777      MOV    #-1, (SP)   ;;SET THE RUBOUT KEY
11940 046516 005303          DEC    R3          ;;BACKUP BY ONE
11941 046520 020327 046700 6$:      CMP    R3, $STTYIN ;;STACK EMPTY?
11942 046524 103434          BLO   4$          ;;BR IF YES
11943 046526 111337 046676      MOVB  (R3), 9$    ;;SETUP TO TYPEOUT THE DELETED CHAR.
11944 046532 104401 046676      TYPE  9$          ;;GO TYPE
11945 046536 000746          BR    2$          ;;GO READ ANOTHER CHAR.
11946 046540 005716          5$:      TST   (SP)        ;;RUBOUT KEY SET?
11947 046542 001406          BEQ   7$          ;;BR IF NO
11948 046544 112737 000134 046676  MOVB   #' \, 9$    ;;TYPE A BACK SLASH
11949 046552 104401 046676      TYPE  9$
11950 046556 005016          CLR   (SP)        ;;CLEAR THE RUBOUT KEY
11951 046560 122713 000025 7$:      CMPB  #25, (R3)   ;;IS CHARACTER A CTRL U?
11952 046564 001003          BNE   8$          ;;BR IF NO
11953 046566 104401 046715      TYPE  $CNTLU      ;;TYPE A CONTROL "U"
11954 046572 000726          BR    1$          ;;GO START OVER
11955 046574 122713 000022 8$:      CMPB  #22, (R3)   ;;IS CHARACTER A "r"?
11956 046600 001011          BNE   3$          ;;BRANCH IF NO
11957 046602 105013          CLRB  (R3)        ;;CLEAR THE CHARACTER
11958 046604 104401 001273      TYPE  , $CRLF     ;;TYPE A "CR" & "LF"
11959 046610 104401 046700      TYPE  $TTYIN     ;;TYPE THE INPUT STRING
11960 046614 000717          BR    2$          ;;GO PICKUP ANOTHER CHACTER
11961 046616 104401 001272 4$:      TYPE  $QUES      ;;TYPE A '?'
11962 046622 000712          BR    1$          ;;CLEAR THE BUFFER AND LOOP
11963 046624 111337 046676 3$:      MOVB  (R3), 9$    ;;ECHO THE CHARACTER
11964 046630 104401 046676      TYPE  9$
11965 046634 122723 000015      CMPB  #15, (R3)+  ;;CHECK FOR RETURN
11966 046640 001305          BNE   2$          ;;LOOP IF NOT RETURN
11967 046642 105063 177777      CLRB  -1(R3)     ;;CLEAR RETURN (THE 15)
11968 046646 104401 001274      TYPE  $LF        ;;TYPE A LINE FEED
11969 046652 005726          TST   (SP)+      ;;CLEAN RUBOUT KEY FROM THE STACK
11970 046654 012603          MOV   (SP)+, R3  ;;RESTORE R3
11971 046656 011646          MOV   (SP), -(SP) ;;ADJUST THE STACK AND PUT ADDRESS OF THE
11972 046660 016666 000004 000002  MOV    4(SP), 2(SP) ;;FIRST ASCII CHARACTER ON IT
11973 046666 012766 046700 000004  MOV    $TTYIN, 4(SP)
11974 046674 000002          RTI
11975 046676 000          9$:      .BYTE 0          ;;RETURN
11976 046677 000          .BYTE 0          ;;STORAGE FOR ASCII CHAR. TO TYPE
11977 046700 000010          $TTYIN: .BLKB 8.  ;;TERMINATOR
11978 046710 041536 005015 000      $CNTLC: .ASCIZ /?C/<15><12> ;;RESERVE 8 BYTES FOR TTY INPUT
11979 046715 136 006525 000012  $CNTLU: .ASCIZ /?U/<15><12> ;;CONTROL "C"
11980 046722 043536 005015 000      $CNTLG: .ASCIZ /?G/<15><12> ;;CONTROL "U"
11981 046727 015 051412 051127  $MSWR:  .ASCIZ <15><12>/SWR = / ;;CONTROL "G"
11982 046734 036440 000040          $MNEW: .ASCIZ / NEW = /
11983 046740 020040 042516 020127
11984 046746 020075 000
11985 046752
11986
11987
11988
11989
11990
11991

```

.EVEN
.SBTTL READ AN OCTAL NUMBER FROM THE TTY

*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
*CHANGE IT TO BINARY.
*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL

```

11992 ;#OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11993 ;#FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11994 ;#THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11995 ;#CALL:
11996 ;# RDOCT ;: READ AN OCTAL NUMBER
11997 ;# RETURN HERE ;: LOW ORDER BITS ARE ON TOP OF THE STACK
11998 ;# ;: HIGH ORDER BITS ARE IN $HIOCT
11999
12000 046752 011646 000004 000002 $RDOCT: MOV (SP),-(SP) ;: PROVIDE SPACE FOR THE
12001 046754 016666 MOV 4(SP),2(SP) ;: INPUT NUMBER
12002 046762 010046 MOV RO,-(SP) ;: PUSH RO ON STACK
12003 046764 010146 MOV R1,-(SP) ;: PUSH R1 ON STACK
12004 046766 010246 MOV R2,-(SP) ;: PUSH R2 ON STACK
12005 046770 104411 1$: RDLIN ;: READ AN ASCII LINE
12006 046772 012600 MOV (SP)+,RO ;: GET ADDRESS OF 1ST CHARACTER
12007 046774 010037 047100 MOV RO,$$ ;: AND SAVE IT
12008 047000 005001 CLR R1 ;: CLEAR DATA WORD
12009 047002 005002 CLR R2
12010 047004 112046 2$: MOV (RO)+,-(SP) ;: PICKUP THIS CHARACTER
12011 047006 001420 BEQ 3$ ;: IF ZERO GET OUT
12012 047010 122716 000060 CMPB #'0,(SP) ;: MAKE SURE THIS CHARACTER
12013 047014 003026 BGT 4$ ;: IS AN OCTAL DIGIT
12014 047016 122716 000067 CMPB #'7,(SP)
12015 047022 002423 BLT 4$
12016 047024 006301 ASL R1 ;: *2
12017 047026 006102 ROL R2 ;: *4
12018 047030 006301 ASL R1 ;: *8
12019 047032 006102 ROL R2
12020 047034 006301 ASL R1
12021 047036 006102 ROL R2
12022 047040 042716 177770 BIC #'C7,(SP) ;: STRIP THE ASCII JUNK
12023 047044 062601 ADD (SP)+,R1 ;: ADD IN THIS DIGIT
12024 047046 000756 BR 2$ ;: LOOP
12025 047050 005726 3$: TST (SP)+ ;: CLEAN TERMINATOR FROM STACK
12026 047052 010166 000012 MOV R1,12(SP) ;: SAVE THE RESULT
12027 047056 010237 047110 MOV R2,$HIOCT
12028 047062 012602 MOV (SP)+,R2 ;: POP STACK INTO R2
12029 047064 012601 MOV (SP)+,R1 ;: POP STACK INTO R1
12030 047066 012600 MOV (SP)+,RO ;: POP STACK INTO RO
12031 047070 000002 RTI ;: RETURN
12032 047072 005726 4$: TST (SP)+ ;: CLEAN PARTIAL FROM STACK
12033 047074 105010 CLR (RO) ;: SET A TERMINATOR
12034 047076 104401 TYPE ;: TYPE UP THRU THE BAD CHAR.
12035 047100 000000 5$: .WORD 0
12036 047102 104401 001272 TYPE $QUES ;: "?" "CR" & "LF"
12037 047106 000730 BR 1$ ;: TRY AGAIN
12038 047110 000000 $HIOCT: .WORD 0 ;: HIGH ORDER BITS GO HERE
12039 .SBTTL SAVE AND RESTORE RO-R5 ROUTINES
12040
12041 ;: *****
12042 ;#SAVE RO-R5
12043 ;#CALL:
12044 ;# SAVREG
12045 ;#UPON RETURN FROM $$SAVREG THE STACK WILL LOOK LIKE:
12046 ;#
12047 ;#TOP---(+16)

```

12048
12049
12050
12051
12052
12053
12054
12055
12056 047112
12057 047112 010046
12058 047114 010146
12059 047116 010246
12060 047120 010346
12061 047122 010446
12062 047124 010546
12063 047126 016646 000022
12064 047132 016646 000022
12065 047136 016646 000022
12066 047142 016646 000022
12067 047146 000002
12068
12069
12070
12071
12072 047150
12073 047150 012666 000022
12074 047154 012666 000022
12075 047160 012666 000022
12076 047164 012666 000022
12077 047170 012605
12078 047172 012604
12079 047174 012603
12080 047176 012602
12081 047200 012601
12082 047202 012600
12083 047204 000002
12084
12085
12086
12087
12088 047206 012737 047346 000024
12089 047214 012737 000340 000026
12090 047222 010046
12091 047224 010146
12092 047226 010246
12093 047230 010346
12094 047232 010446
12095 047234 010546
12096 047236 017746 131676
12097 047242 010637 047352
12098 047246 012737 047260 000024
12099 047254 000000
12100 047256 000776
12101
12102
12103

```

;* +2---(+18)
;* +4---R5
;* +6---R4
;* +8---R3
;* +10---R2
;* +12---R1
;* +14---R0

```

```

$SAVREG:
MOV RO,-(SP)      ;; PUSH R0 ON STACK
MOV R1,-(SP)      ;; PUSH R1 ON STACK
MOV R2,-(SP)      ;; PUSH R2 ON STACK
MOV R3,-(SP)      ;; PUSH R3 ON STACK
MOV R4,-(SP)      ;; PUSH R4 ON STACK
MOV R5,-(SP)      ;; PUSH R5 ON STACK
MOV 22(SP),-(SP)  ;; SAVE PS OF MAIN FLOW
MOV 22(SP),-(SP)  ;; SAVE PC OF MAIN FLOW
MOV 22(SP),-(SP)  ;; SAVE PS OF CALL
MOV 22(SP),-(SP)  ;; SAVE PC OF CALL
RTI

```

```

;*RESTORE RO-R5
;*CALL:
;* RESREG

```

```

$RESREG:
MOV (SP)+,22(SP)  ;; RESTORE PC OF CALL
MOV (SP)+,22(SP)  ;; RESTORE PS OF CALL
MOV (SP)+,22(SP)  ;; RESTORE PC OF MAIN FLOW
MOV (SP)+,22(SP)  ;; RESTORE PS OF MAIN FLOW
MOV (SP)+,R5      ;; POP STACK INTO R5
MOV (SP)+,R4      ;; POP STACK INTO R4
MOV (SP)+,R3      ;; POP STACK INTO R3
MOV (SP)+,R2      ;; POP STACK INTO R2
MOV (SP)+,R1      ;; POP STACK INTO R1
MOV (SP)+,R0      ;; POP STACK INTO R0
RTI

```

.SBTTL POWER DOWN AND UP ROUTINES

```

;*****
;POWER DOWN ROUTINE
$PWRDN: MOV #SILLUP,@#PWRVEC ;; .SET FOR FAST UP
MOV #340,@#PWRVEC+2 ;; PRIO:7
MOV RO,-(SP)      ;; PUSH R0 ON STACK
MOV R1,-(SP)      ;; PUSH R1 ON STACK
MOV R2,-(SP)      ;; PUSH R2 ON STACK
MOV R3,-(SP)      ;; PUSH R3 ON STACK
MOV R4,-(SP)      ;; PUSH R4 ON STACK
MOV R5,-(SP)      ;; PUSH R5 ON STACK
MOV @SWR,-(SP)    ;; PUSH @SWR ON STACK
MOV SP,$SAVR6     ;; SAVE SP
MOV #SPWRUP,@#PWRVEC ;; SET UP VECTOR
HALT
BR -.2           ;; HANG UP

```

```

;*****
;POWER UP ROUTINE

```



```

12104 047260 012737 047346 000024 $PWRUP: MOV $SILLUP,2#$PWRVEC ;;SET FOR FAST DOWN
12105 047266 013706 047352 MOV $SAVR6,SP ;;GET SP
12106 047272 005037 047352 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
12107 047276 005237 047352 1$: INC $SAVR6 ;;WAIT FOR THE INC
12108 047302 001375 BNE 1$ ;;OF WORD
12109 047304 012677 131630 MOV (SP)+,2$SWR ;;POP STACK INTO 2$SWR
12110 047310 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
12111 047312 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
12112 047314 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
12113 047316 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
12114 047320 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
12115 047322 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
12116 047324 012737 047206 000024 MOV $PWRDN,2#$PWRVEC ;;SET UP THE POWER DOWN VECTOR
12117 047332 012737 000340 000026 MOV #340,2#$PWRVEC+2 ;;PRIO:7
12118 047340 104401 TYPE ;;REPORT THE POWER FAILURE
12119 047342 047354 $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
12120 047344 000002 RTI
12121 047346 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
12122 047350 000776 BR .-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
12123 047352 000000 $SAVR6: 0 ;;PUT THE SP HERE
12124 047354 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
12125 047362 000122
12126
12127 .SBTTL EVEN
12128 TRAP DECODER
12129
12130 ;*****
12131 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
12132 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
12133 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
12134 ;*GO TO THAT ROUTINE.
12135
12135 047364 010046 $STRAP: MOV R0,-(SP) ;;SAVE R0
12136 047366 016600 000002 MOV 2(SP),R0 ;;GET TRAP ADDRESS
12137 047372 005740 TST -(R0) ;;BACKUP BY 2
12138 047374 111000 MOVB (R0),R0 ;;GET RIGHT BYTE OF TRAP
12139 047376 006300 ASL R0 ;;POSITION FOR INDEXING
12140 047400 016000 047420 MOV $TRPAD(R0),R0 ;;INDEX TO TABLE
12141 047404 000200 RTS R0 ;;GO TO ROUTINE
12142
12143
12144 ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
12145
12146 047406 011646 $STRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
12147 047410 016666 000004 000002 MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
12148 047416 000002 RTI ;;RESTORE THE PSW
12149
12150 .SBTTL TRAP TABLE
12151
12152 ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
12153 ;*BY THE "TRAP" INSTRUCTION.
12154
12155 ; ROUTINE
12156 ; -----
12157 047420 047406 $TRPAD: .WORD $STRAP2
12158 047422 044560 $TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
12159 047424 045312 $TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)

```

E02

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 224
TRAP TABLE

SEQ 0224

Address	Octal 1	Octal 2	Call Name	Trap #	Description
12160	047426	045266	\$TYPOS	TRAP+3(104403)	TYPE OCTAL NUMBER (NO LEADING ZEROS)
12161	047430	045326	\$TYPON	TRAP+4(104404)	TYPE OCTAL NUMBER (AS PER LAST CALL)
12162	047432	045042	\$TYPDS	TRAP+5(104405)	TYPE DECIMAL NUMBER (WITH SIGN)
12163					
12164	047434	046102	\$GTSWR	TRAP+6(104406)	GET SOFT-SWR SETTING
12165					
12166	047436	046012	\$CKSWR	TRAP+7(104407)	TEST FOR CHANGE IN SOFT-SWR
12167	047440	046354	\$RDCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE
12168	047442	046444	\$RDLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE
12169	047444	046752	\$RDOCT	TRAP+12(104412)	READ AN OCTAL NUMBER FROM TTY
12170	047446	047112	\$SAVREG	TRAP+13(104413)	SAVE RO-RS ROUTINE
12171	047450	047150	\$RESREG	TRAP+14(104414)	RESTORE RO-RS ROUTINE
12172	047452	035600	\$SCOP1\$	TRAP+15(104415)	INTERNAL LOOP ON ERROR
12173	047454	037554	\$SSINIT	TRAP+16(104416)	INITIALIZE SUBSYSTEM
12174	047456	036312	\$LOADRK	TRAP+17(104417)	LOAD RK611 FOR OPERATION
12175	047460	036542	\$GETRK	TRAP+20(104420)	GET RK611 REGISTERS
12176	047462	040540	\$CHKOP	TRAP+21(104421)	CHECK OPERATION FOR ANY ERRORS
12177	047464	040506	\$CHKWE	TRAP+22(104422)	CHECK OPERATION FOR EXPECTED ERRORS
12178	047466	036106	\$IWAT16	TRAP+23(104423)	WAIT 16 MS
12179	047470	036076	\$IWAT32	TRAP+24(104424)	WAIT 32 MS
12180	047472	036066	\$IWAT48	TRAP+25(104425)	WAIT 48 MS
12181	047474	036056	\$IWAT64	TRAP+26(104426)	WAIT 64 MS
12182	047476	036046	\$IWAT80	TRAP+27(104427)	WAIT 80 MS
12183	047500	036036	\$IWAT96	TRAP+30(104430)	WAIT 96 MS
12184	047502	036026	\$IWAT112	TRAP+31(104431)	WAIT 112 MS
12185	047504	036016	\$IWAT128	TRAP+32(104432)	WAIT 128 MS
12186	047506	036006	\$IWAT144	TRAP+33(104433)	WAIT 144 MS
12187	047510	035776	\$IWAT159	TRAP+34(104434)	WAIT 160 MS
12188	047512	035766	\$IWAT15	TRAP+35(104435)	WAIT FOR 1 SECOND
12189	047514	035756	\$IWAT25	TRAP+36(104436)	WAIT FOR 2 SECONDS
12190	047516	035736	\$IWAT85	TRAP+37(104437)	WAIT FOR 8 SECONDS
12191	047520	035746	\$IWAT1M	TRAP+40(104440)	WAIT FOR 1 MIN
12192		000102	\$TERM=-\$TRPAD		

DATA PATTERNS

.SBTTL DATA PATTERNS
;DATA PATTERN 1
; PATTERN IS ALL ZEROS

;DATA PATTERN 2
; HI-LO FREQ. MIX

PAT02: 177777
177777
177777
052525
052525
052525
177777
177777
052525
052525
177777
052525
177252
177252
172765
172765

;DATA PATTERN 3
; HI FREQ. PHASE MIX

PAT03: 000000
000000
000000
177777
177777
177777
000000
000000
177777
177777
000000
000000
177777
000000
177777
000000
177777

;DATA PATTERN 4
; LO FREQ. PHASE MIX

PAT04: 052525
052525
052525
125252
125252
125252
052525
052525
125252

12193
12194
12195
12196
12197
12198
12199 047522
12200 047522 177777
12201 047524 177777
12202 047526 177777
12203 047530 052525
12204 047532 052525
12205 047534 052525
12206 047536 177777
12207 047540 177777
12208 047542 052525
12209 047544 052525
12210 047546 177777
12211 047550 052525
12212 047552 177252
12213 047554 177252
12214 047556 172765
12215 047560 172765
12216
12217
12218
12219 047562
12220 047562 000000
12221 047564 000000
12222 047566 000000
12223 047570 177777
12224 047572 177777
12225 047574 177777
12226 047576 000000
12227 047600 000000
12228 047602 177777
12229 047604 177777
12230 047606 000000
12231 047610 177777
12232 047612 000000
12233 047614 177777
12234 047616 000000
12235 047620 177777
12236
12237
12238
12239 047622
12240 047622 052525
12241 047624 052525
12242 047626 052525
12243 047630 125252
12244 047632 125252
12245 047634 125252
12246 047636 052525
12247 047640 052525
12248 047642 125252

12249	047644	125252	125252
12250	047646	052525	052525
12251	047650	125252	125252
12252	047652	052525	052525
12253	047654	125252	125252
12254	047656	052525	052525
12255	047660	125252	125252

;DATA PATTERN 5
; MAX PRECOMP. PHASE MIX

PAT05:

12259	047662	133333	133333
12260	047662	066666	066666
12261	047664	155555	155555
12262	047666	155555	155555
12263	047670	155555	155555
12264	047672	133333	133333
12265	047674	066666	066666
12266	047676	066666	066666
12267	047700	155555	155555
12268	047702	155555	155555
12269	047704	133333	133333
12270	047706	133333	133333
12271	047710	133333	133333
12272	047712	133333	133333
12273	047714	133333	133333
12274	047716	133333	133333
12275	047720	133333	133333

;DATA PATTERN 6
; ROTATING BOUNDARY PULSE PRECOMP.

PAT06:

12279	047722	121105	121105
12280	047722	150442	150442
12281	047724	064221	064221
12282	047726	132110	132110
12283	047730	055044	055044
12284	047732	026422	026422
12285	047734	013211	013211
12286	047736	105504	105504
12287	047740	042642	042642
12288	047742	021321	021321
12289	047744	110550	110550
12290	047746	044264	044264
12291	047750	022132	022132
12292	047752	011055	011055
12293	047754	104426	104426
12294	047756	042213	042213

;DATA PATTERN 7
; FIELD OF ALL ONES

;DATA PATTERN 10
; ROTATING CELL PULSE PRECOMP.

PAT10:

12302	047762	026455	026455
12303	047762	113226	113226
12304	047764		

12276			
12277			
12278			
12279	047722		
12280	047722	121105	
12281	047724	150442	
12282	047726	064221	
12283	047730	132110	
12284	047732	055044	
12285	047734	026422	
12286	047736	013211	
12287	047740	105504	
12288	047742	042642	
12289	047744	021321	
12290	047746	110550	
12291	047750	044264	
12292	047752	022132	
12293	047754	011055	
12294	047756	104426	
12295	047760	042213	
12296			
12297			
12298			
12299			
12300			
12301			
12302	047762		
12303	047762	026455	
12304	047764	113226	

12305	047766	045513	045513
12306	047770	122645	122645
12307	047772	151322	151322
12308	047774	064551	064551
12309	047776	132264	132264
12310	050000	055132	055132
12311	050002	026455	026455
12312	050004	113226	113226
12313	050006	045513	045513
12314	050010	122645	122645
12315	050012	151322	151322
12316	050014	064551	064551
12317	050016	132264	132264
12318	050020	055132	055132
12319			
12320			
12321			

```

;DATA PATTERN 11
;PAT11: SHIFTED 1 IN A FIELD OF ZEROS

```

12322	050022	000001	000001
12323	050024	000002	000002
12324	050026	000004	000004
12325	050030	000010	000010
12326	050032	000020	000020
12327	050034	000040	000040
12328	050036	000100	000100
12329	050040	000200	000200
12330	050042	000400	000400
12331	050044	001000	001000
12332	050046	002000	002000
12333	050050	004000	004000
12334	050052	010000	010000
12335	050054	020000	020000
12336	050056	040000	040000
12337	050060	100000	100000
12338			
12339			
12340			
12341			

```

;DATA PATTERN 12
;PAT12: SHIFTED 0 IN A FIELD OF ONES

```

12342	050062	177776	177776
12343	050064	177775	177775
12344	050066	177773	177773
12345	050070	177767	177767
12346	050072	177757	177757
12347	050074	177737	177737
12348	050076	177677	177677
12349	050100	177577	177577
12350	050102	177377	177377
12351	050104	176777	176777
12352	050106	175777	175777
12353	050110	173777	173777
12354	050112	167777	167777
12355	050114	157777	157777
12356	050116	137777	137777
12357	050120	077777	077777
12358			
12359			
12360			

```

;DATA PATTERN 13

```

12361
12362 050122
12363 050122 052525
12364 050124 052525
12365 050126 052525
12366 050130 052525
12367 050132 052525
12368 050134 052525
12369 050136 052525
12370 050140 052525
12371 050142 052525
12372 050144 052525
12373 050146 052525
12374 050150 052525
12375 050152 052525
12376 050154 052525
12377 050156 052525
12378 050160 052525
12379
12380
12381
12382 050162
12383 050162 125252
12384 050164 125252
12385 050166 125252
12386 050170 125252
12387 050172 125252
12388 050174 125252
12389 050176 125252
12390 050200 125252
12391 050202 125252
12392 050204 125252
12393 050206 125252
12394 050210 125252
12395 050212 125252
12396 050214 125252
12397 050216 125252
12398 050220 125252
12399
12400
12401
12402 050222
12403 050222 000001
12404 050224 000003
12405 050226 000007
12406 050230 000017
12407 050232 000037
12408 050234 000077
12409 050236 000177
12410 050240 000377
12411 050242 000777
12412 050244 001777
12413 050246 003777
12414 050250 007777
12415 050252 017777
12416 050254 037777

PAT13: ALTERNATING 0-1

052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525
052525

;DATA PATTERN 14
PAT14: ALTERNATING 1-0

125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252
125252

;DATA PATTERN 15
PAT15: SHIFTING ZEROS AND ONES

000001
000003
000007
000017
000037
000077
000177
000377
000777
001777
003777
007777
017777
037777

12417	050256	077777
12418	050260	177777
12419		
12420		
12421		
12422	050262	
12423	050262	072307
12424	050264	135143
12425	050266	156461
12426	050270	167230
12427	050272	073514
12428	050274	035646
12429	050276	016723
12430	050300	107351
12431	050302	143564
12432	050304	061672
12433	050306	030735
12434	050310	114356
12435	050312	046167
12436	050314	123073
12437	050316	151453
12438	050320	164616
12439		

077777
177777

:DATA PATTERN 16
: PAT16: COMPOSITE ROTATING

072307
135143
156461
167230
073514
035646
016723
107351
143564
061672
030735
114356
046167
123073
151453
164616

12440		.SBTTL	FIELDS AND VARIABLES FOR OPERATION CHECKING	
12441	024000	CS1ERBIT=24000		;CSI ERROR BITS SPAR & CTO
12442	177400	CS2ERBIT=177400		;CS2 ERROR BITS
12443				;DLT,WCE,UPE,NED,NEM
12444				;PGE,MOS,UFE
12445	000070	DSERBIT=70		;DS ERROR BITS
12446				;SPDLSS,DROT,ACLO
12447				
12448	050322	000000	EXPWC: .WORD 0	;EXPECTED WORD COUNT (GIVEN)
12449	050324	000000	EXPUBA: .WORD 0	;EXPECTED UPPER BA (COMPUTED)
12450	050326	000000	EXPBA: .WORD 0	;EXPECTED BUS ADDRESS (COMPUTED)
12451	050330	000000	EXPCYL: .WORD 0	;EXPECTED CYLINDER (COMPUTED)
12452	050332	000000	EXPSEC: .WORD 0	;EXPECTED SECTOR (COMPUTED)
12453	050334	000000	EXPTRK: .WORD 0	;EXPECTED TRACK (COMPUTED)
12454				
12455	050336	000000	REALWC: .WORD 0	;WORD COUNT AT END OF OPERATION
12456	050340	000000	REALUB: .WORD 0	;REAL UPPER BA
12457	050342	000000	REALBA: .WORD 0	;BUS ADDRESS
12458	050344	000000	REALCY: .WORD 0	;CYLINDER
12459	050346	000000	REALTRK: .WORD 0	;TRACK
12460	050350	000000	REALSEC: .WORD 0	;SECTOR
12461				
12462	050352	000000	GRP1ER: .WORD 0	;GROUP 1 ERROR FIELDS
12463		000001	SPARERR=BIT0	;CONTROLLER DETECTED DRIVE BUS PARITY ERR
12464		000002	SKIERR=BIT1	;SEEK INCOMPLETE
12465		000004	NXFERR=BIT2	;NON-EXECUTABLE DRIVE FUNCTION
12466		000010	DRPARERR=BIT3	;DRIVE DETECTED DRIVE BUS PARITY ERROR
12467		000020	FMTERR=BIT4	;FORMAT ERROR
12468		000040	DTYERR=BIT5	;DRIVE TYPE ERROR
12469		000100	ACLOERR=BIT6	;AC LOW ERROR
12470		000200	SPDERR=BIT7	;SPEED LOSS ERROR
12471		000400	DROTERR=BIT8	;DRIVE OFF TRACK ERROR
12472		001000	COERR=BIT9	;CYLINDER OVER FLOW ERROR
12473		002000	IDAERR=BIT10	;ILLEGAL DISK ADDRESS ERROR
12474		004000	WLERR=BIT11	;WRITE LOCK ERROR
12475		010000	DTERR=BIT12	;DRIVE TIMING ERROR
12476		020000	NCERWE=BIT13	;NO CERR WITH ERROR SET ERROR
12477		040000	UNSERR=BIT14	;DRIVE UNSAFE ERROR
12478		100000	CERNER=BIT15	;CERR BUT NO ERROR SET ERROR
12479				
12480	050354	000000	GRP2ER: .WORD 0	;GROUP 2 ERROR FIELD
12481		000001	ECHERR=BIT0	;ECC HARD ERROR
12482		000002	DCKERR=BIT1	;DATA CHECK ERROR
12483		000004	WCKERR=BIT2	;WRITE CHECK ERROR
12484		000010	DLTERR=BIT3	;DATA LATE ERROR
12485		000020	OPIERR=BIT4	;OPERATION INCOMPLETE ERROR
12486		000040	HVRCERR=BIT5	;HEADER VRC ERROR
12487		000100	BSERR=BIT6	;BAD SECTOR ERROR
12488				
12489	050356	000000	GRP3ER: .WORD 0	;GROUP 3 ERROR FIELD
12490		000001	NEDERR=BIT0	;NON-EXISTANT DRIVE ERROR
12491		000002	CTOERR=BIT1	;CONTROLLER TIME OUT ERROR
12492		000004	UFERR=BIT2	;UNIT FIELD ERROR
12493		000010	MDSERR=BIT3	;MULTIPLE DRIVE SELECT ERROR
12494		000020	PGERR=BIT4	;PROGRAMMING ERROR
12495		000040	NEMERR=BIT5	;NON-EXISTANT MEMORY ERROR

12496		000100	UPERR=	BIT6	;UNIBUS PARITY ERROR
12497		000200	ILFERR=	BIT7	;ILLEGAL FUNCTION ERROR.
12498					
12499	050360	000000	GRP4ER:	0	;GROUP 4 ERROR FIELD
12500		000001	WCERR=	BIT0	;WORD COUNT ERROR FLAG
12501		000002	UBAERR=	BIT1	;UPPER BA ERROR
12502		000004	BAERR=	BIT2	;BUS ADDRESS ERROR FLAG
12503		000010	CYLERR=	BIT3	;CYL ADDRESS ERROR FLAG
12504		000020	TRKERR=	BIT4	;TRACK ADDRESS ERROR FLAG
12505		000040	SECERR=	BIT5	;SECTOR ADDRESS ERROR FLAG
12506					
12507	050362	054440	GRP4MS:	EM10	
12508	050364	054513		EM11A	
12509	050366	054465		EM11	
12510	050370	054567		EM12	
12511	050372	054622		EM13	
12512	050374	054652		EM14	
12513					
12514	050376	054704	GRP3MS:	EM15	
12515	050400	054727		EM16	
12516	050402	054752		EM17	
12517	050404	054773		EM18	
12518	050406	055020		EM19	
12519	050410	055042		EM20	
12520	050412	055066		EM21	
12521	050414	055112		EM22	
12522					
12523	050416	055133	GRP2MS:	EM23	
12524	050420	055144		EM24	
12525	050422	055157		EM25	
12526	050424	055173		EM26	
12527	050426	055205		EM27	
12528	050430	055232		EM28	
12529	050432	055245		EM29	
12530					
12531	050434	055266	GRP1MS:	EM30	
12532	050436	055341		EM31	
12533	050440	055361		EM32	
12534	050442	055417		EM33	
12535	050444	055465		EM34	
12536	050446	055502		EM35	
12537	050450	055523		EM36	
12538	050452	055532		EM37	
12539	050454	055555		EM38	
12540	050456	055575		EM39	
12541	050460	055617		EM40	
12542	050462	055651		EM41	
12543	050464	055672		EM42	
12544	050466	055715		EM43	
12545	050470	055757		EM44	
12546	050472	055774		EM45	
12547					
12548	050474	000000	GPSUMF:	0	;GROUP ERROR SUMMARY FLAGS
12549		000001	GRP1ST=	BIT0	;GROUP 1 ERROR SET
12550		000002	GRP2ST=	BIT1	;GROUP 2 ERROR SET
12551		000004	GRP3ST=	BIT2	;GROUP 3 ERROR SET

M02

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 232
FIELDS AND VARIABLES FOR OPERATION CHECKING

SEQ 0232

12552	000010	GP1NR=	BIT3	:GROUP 1 ERROR NOT RECEIVED
12553	000020	GP2NR=	BIT4	:GROUP 2 ERROR NOT RECEIVED
12554	000040	GP3NR=	BIT5	:GROUP 3 ERROR NOT RECEIVED
12555	040000	DRSTER=	BIT14	:ERROR IN GETTING DRIVE STATUS FLAG.
12556	100000	REPNR=	BIT15	:REPORTING NOT RECEIVED SWITCH

12557
12558
12559
12560
12561

```

.SBTTL TABLE OF OPERATION MESSAGE ADDRESS
;* THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE
;* USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.
    
```

12562	050476	050536	CMNDLB: .WORD	OPER00	:ADDRESS OF SELECT MESSAGE
12563	050500	050553	.WORD	OPER02	PACK ACK
12564	050502	050564	.WORD	OPER04	DRIVE CLEAR
12565	050504	050600	.WORD	OPER06	UNLOAD
12566	050506	050607	.WORD	OPER10	START SPINDLE
12567	050510	050625	.WORD	OPER12	RECALIBRATE
12568	050512	050641	.WORD	OPER14	OFFSET
12569	050514	050650	.WORD	OPER16	SEEK
12570	050516	050655	.WORD	OPER20	READ DATA
12571	050520	050667	.WORD	OPER22	WRITE DATA
12572	050522	050702	.WORD	OPER24	READ HEADER
12573	050524	050717	.WORD	OPER26	WRITE HEADER
12574	050526	050735	.WORD	OPER30	WRITE CHECK
12575	050530	050751	.WORD	OPER32	:ILLEGAL OPERATION 33
12576	050532	050775	.WORD	OPER34	35
12577	050534	051021	.WORD	OPER36	37

12578	12579	12580	050536	051104	053111	020105	.SBTTL OPERATION MESSAGES
12581	050544	042523	042514	052103	OPER00: .ASCIZ	/DRIVE SELECT/	
12582	050552	000					
12583	050553	120	041501	020113	OPER02: .ASCIZ	/PACK ACK/	
12584	050560	041501	000113				
12585	050564	051104	053111	020105	OPER04: .ASCIZ	/DRIVE CLEAR/	
12586	050572	046103	040505	000122			
12587	050600	047125	047514	042101	OPER06: .ASCIZ	/UNLOAD/	
12588	050606	000					
12589	050607	123	040524	052122	OPER10: .ASCIZ	/START SPINDLE/	
12590	050614	051440	044520	042116			
12591	050622	042514	000				
12592	050625	122	041505	046101	OPER12: .ASCIZ	/RECALIBRATE/	
12593	050632	041111	040522	042524			
12594	050640	000					
12595	050641	117	043106	042523	OPER14: .ASCIZ	/OFFSET/	
12596	050646	000124					
12597	050650	042523	045505	000	OPER16: .ASCIZ	/SEEK/	
12598	050655	122	040505	020104	OPER20: .ASCIZ	/READ DATA/	
12599	050662	040504	040524	000			
12600	050667	127	044522	042524	OPER22: .ASCIZ	/WRITE DATA/	
12601	050674	042040	052101	000101			
12602	050702	042522	042101	044040	OPER24: .ASCIZ	/READ HEADERS/	
12603	050710	040505	042504	051522			
12604	050716	000					
12605	050717	127	044522	042524	OPER26: .ASCIZ	/WRITE HEADERS/	
12606	050724	044040	040505	042504			
12607	050732	051522	000				

12608	050735	127	044522	042524	OPER30: .ASCIZ /WRITE CHECK/
12609	050742	041440	042510	045503	
12610	050750	000			
12611	050751	111	046114	043505	OPER32: .ASCIZ /ILLEGAL FUNCTION 33/
12612	050756	046101	043040	047125	
12613	050764	052103	047511	020116	
12614	050772	031463	000		
12615	050775	111	046114	043505	OPER34: .ASCIZ /ILLEGAL FUNCTION 35/
12616	051002	046101	043040	047125	
12617	051010	052103	047511	020116	
12618	051016	032463	000		
12619	051021	111	046114	043505	OPER36: .ASCIZ /ILLEGAL FUNCTION 37/
12620	051026	046101	043040	047125	
12621	051034	052103	047511	020116	
12622	051042	033463	000		
12623	051045	127	044522	042524	OPER37: .ASCIZ /WRITE DATA ABORTED WITH BSE/
12624	051052	042040	052101	020101	
12625	051060	041101	051117	042524	
12626	051066	020104	044527	044124	
12627	051074	041040	042523	000	
12628	051101				OPER40:
12629	051101	127	044522	042524	OPER41: .ASCIZ /WRITE CHECK ABORTED WITH WCE/
12630	051106	041440	042510	045503	
12631	051114	040440	047502	052122	
12632	051122	042105	053440	052111	
12633	051130	020110	041527	000105	
12634	051136	051127	052111	020105	OPER42: .ASCIZ /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
12635	051144	040504	040524	040440	
12636	051152	047502	052122	042105	
12637	051160	053440	052111	020110	
12638	051166	047516	026516	054105	
12639	051174	051511	040524	052116	
12640	051202	046440	046505	051117	
12641	051210	000131			
12642	051212	042522	042101	042040	OPER43: .ASCIZ /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
12643	051220	052101	020101	041101	
12644	051226	051117	042524	020104	
12645	051234	044527	044124	047040	
12646	051242	047117	042455	044530	
12647	051250	052123	047101	020124	
12648	051256	042515	047515	054522	
12649	051264	000			
12650	051265	127	044522	042524	OPER44: .ASCIZ /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR/
12651	051272	042040	052101	020101	
12652	051300	041101	051117	042524	
12653	051306	020104	044527	044124	
12654	051314	052440	044516	052502	
12655	051322	020123	040520	044522	
12656	051330	054524	042440	051122	
12657	051336	051117	000		
12658					

.SBTTL ASCII MESSAGES

```

12659
12660
12661 051341 040 000040
12662 051344 005015 045522 030466
12663 051352 020061 052502 020123
12664 051360 042101 051104 051505
12665 051366 020123 020050 000
12666 051373 040 020051 020075
12667 051400 000
12668 051401 122 033113 030461
12669 051406 053040 041505 047524
12670 051414 020122 042101 051104
12671 051422 051505 020123 020050
12672 051430 000
12673 051431 122 033113 030461
12674 051436 050040 044522 051117
12675 051444 052111 020131 020050
12676 051452 000
12677 051453 111 051516 043125
12678 051460 044506 044503 047105
12679 051466 020124 042515 047515
12680 051474 054522 020056 051120
12681 051502 043517 040522 020115
12682 051510 041101 051117 044524
12683 051516 043516 006456 000012
12684 051524
12685
12686
12687 051524
12688 051524 005015 044103 047101
12689 051532 042507 054040 042130
12690 051540 020120 040520 045503
12691 051546 005015 046103 040505
12692 051554 020122 047514 020103
12693 051562 030064 051054 051505
12694 051570 040524 052122 050040
12695 051576 047522 051107 046501
12696 051604 000
12697
12698
12699 051605 015 047012 020117
12700 051612 051104 053111 051505
12701 051620 040440 040526 046111
12702 051626 041101 042514 043040
12703 051634 051117 052040 051505
12704 051642 044524 043516 020056
12705 051650 051120 043517 040522
12706 051656 020115 041101 051117
12707 051664 042524 000104
12708 051670 005015 044124 020105
12709 051676 047506 046114 053517
12710 051704 047111 020107 051104
12711 051712 053111 051505 053440
12712 051720 046111 020114 042502
12713 051726 052040 051505 042524
12714 051734 006504 000012

```

```

SPACE2: .ASCIZ / /
OPR001: .ASCIZ <15><12>/RK611 BUS ADDRESS ( /

OPR002: .ASCIZ / ) = /

OPR003: .ASCIZ /RK611 VECTOR ADDRESS ( /

OPR004: .ASCIZ /RK611 PRIORITY ( /

OPR005: .ASCIZ /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>

OPR006:
: .ASCII <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/
: .ASCIZ <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/
OPR007: .ASCII <CR><LF>/CHANGE XXDP PACK/

: .ASCIZ <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/

: .ASCII <15><12>/DRIVE 0 WILL NOT BE TESTED. TO TEST DRIVE 0,/<15><12>
: .ASCIZ /RESTART AT 214 AND MOUNT SCRATCH PACK AS DIRECTED./
OPR008: .ASCIZ <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/

OPR009: .ASCIZ <15><12>/THE FOLLOWING DRIVES WILL BE TESTED/<15><12>

```

12715	051740	005015	047516	050040	OPR010: .ASCII <15><12>/NO PARITY OPTION FOR MEMORY AREA IN USE/<15><12>
12716	051746	051101	052111	020131	
12717	051754	050117	044524	047117	
12718	051762	043040	051117	046440	
12719	051770	046505	051117	020131	
12720	051776	051101	040505	044440	
12721	052004	020116	051525	006505	
12722	052012	012			
12723	052013	106	051117	052040	.ASCII /FOR TEST - UNIBUS PARITY ERROR AND/<15><12>
12724	052020	051505	020124	020055	
12725	052026	047125	041111	051525	
12726	052034	050040	051101	052111	
12727	052042	020131	051105	047522	
12728	052050	020122	047101	006504	
12729	052056	012			
12730	052057	062	024064	024470	.ASCIZ /24(8) SECTOR FORMAT DATA XFER TEST BYPASSED/<15><12>
12731	052064	051440	041505	047524	
12732	052072	020122	047506	046522	
12733	052100	052101	042040	052101	
12734	052106	020101	043130	051105	
12735	052114	052040	051505	020124	
12736	052122	054502	040520	051523	
12737	052130	042105	005015	000	
12738	052135	015	046412	046505	OPR011: .ASCII <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS
12739	052142	051117	020131	044523	
12740	052150	042532	047040	052117	
12741	052156	046040	051101	042507	
12742	052164	042440	047516	043525	
12743	052172	020110	047506	020122	
12744	052200	052502	020123	042101	
12745	052206	051104	051505	020123	
12746	052214	044502	051524	030440	
12747	052222	020066	020046	033461	
12748	052230	052040	051505	051524	
12749	052236	005015			
12750	052240	046101	020114	040504	.ASCIZ /ALL DATA XFER TESTS WITH ADDR >/
12751	052246	040524	054040	042506	
12752	052254	020122	042524	052123	
12753	052262	020123	044527	044124	
12754	052270	040440	042104	020122	
12755	052276	000076			
12756	052300	031063	000113		OPR012: .ASCIZ /32K/
12757	052304	032066	000113		OPR013: .ASCIZ /64K/
12758	052310	033071	000113		OPR014: .ASCIZ /96K/
12759	052314	041040	050131	051501	OPR015: .ASCIZ / BYPASSED/<15><12>
12760	052322	042523	006504	000012	
12761	052330	005015	020012	020040	OPR016: .ASCII <15><12><12>/ *** CAUTION ***/<15><12><12>
12762	052336	020040	020040	025052	
12763	052344	020052	040503	052125	
12764	052352	047511	020116	025052	
12765	052360	006452	005012		
12766	052364	044124	051511	050040	.ASCII /THIS PROGRAM SHOULD BE HALTED BY TYPING ↑C./<15><12>
12767	052372	047522	051107	046501	
12768	052400	051440	047510	046125	
12769	052406	020104	042502	044040	
12770	052414	046101	042524	020104	

12771	052422	054502	052040	050131
12772	052430	047111	020107	041536
12773	052436	006456	012	
12774	052441	111	020106	040510
12775	052446	052114	042105	052440
12776	052454	044523	043516	052040
12777	052462	042510	044040	046101
12778	052470	020124	042513	026131
12779	052476	052040	042510	051440
12780	052504	040524	042524	047440
12781	052512	020106	044124	020105
12782	052520	051104	053111	006505
12783	052526	012		
12784	052527	117	020122	040503
12785	052534	052122	044522	043504
12786	052542	020105	040503	047116
12787	052550	052117	041040	020105
12788	052556	051120	042105	041511
12789	052564	042524	027104	005015
12790	052572	012		
12791	052573	101	046114	042040
12792	052600	044522	042526	020123
12793	052606	047524	041040	020105
12794	052614	042524	052123	042105
12795	052622	046440	051525	020124
12796	052630	042502	047440	026516
12797	052636	044514	042516	006454
12798	052644	012		
12799	052645	122	040505	054504
12800	052652	020054	047101	020104
12801	052660	051127	052111	020105
12802	052666	047514	045503	051040
12803	052674	051505	052105	006456
12804	052702	012		
12805	052703	101	054516	042040
12806	052710	044522	042526	047040
12807	052716	052117	052040	020117
12808	052724	042502	052040	051505
12809	052732	042524	020104	052515
12810	052740	052123	041040	020105
12811	052746	043117	026506	044514
12812	052754	042516	006456	005012
12813	052762	047516	042524	020072
12814	052770	047062	020104	047101
12815	052776	020104	052523	051502
12816	053004	050505	042525	052116
12817	053012	050040	051501	020123
12818	053020	052522	020116	044524
12819	053026	042515	044440	006523
12820	053034	012		
12821	053035	040	020040	020040
12822	053042	040440	050120	047522
12823	053050	020130	020062	044515
12824	053056	020116	030063	051440
12825	053064	041505	043040	051117
12826	053072	042440	041501	020110

.ASCII /IF HALTED USING THE HALT KEY, THE STATE OF THE DRIVE/<15><12>

.ASCII /OR CARTRIDGE CANNOT BE PREDICTED./<15><12><12>

.ASCII /ALL DRIVES TO BE TESTED MUST BE ON-LINE,/<15><12>

.ASCII /READY, AND WRITE LOCK RESET./<15><12>

.ASCII /ANY DRIVE NOT TO BE TESTED MUST BE OFF-LINE./<15><12><12>

.ASCII /NOTE: 2ND AND SUBSEQUENT PASS RUN TIME IS/<15><12>

.ASCIZ / APPROX 2 MIN 30 SEC FOR EACH DRIVE./<15><12>

E03

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 237
ASCII MESSAGES

SEQ 0237

12827	053100	051104	053111	027105	
12828	053106	005015	000		
12829	053111	015	043012	051111	OPR017: .ASCII <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./
12830	053116	052123	031040	033065	
12831	053124	051440	041505	047524	
12832	053132	051522	047040	052117	
12833	053140	041040	042523	042440	
12834	053146	051122	051117	043040	
12835	053154	042522	027105		
12836	053160	040515	044530	052515	.ASCIZ /MAXIMUM DATA TRANSFER TEST BYPASSED/<15><12>
12837	053166	020115	040504	040524	
12838	053174	052040	040522	051516	
12839	053202	042506	020122	042524	
12840	053210	052123	041040	050131	
12841	053216	051501	042523	006504	
12842	053224	000012			
12843	053226	020040	020040	006440	OPR018: .ASCIZ / /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>
12844	053234	047412	051116	020131	
12845	053242	020061	051104	053111	
12846	053250	027105	07440	042526	
12847	053256	046122	050101	042520	
12848	053264	020104	050117	051105	
12849	053272	052101	047511	051516	
12850	053300	041040	050131	051501	
12851	053306	042523	006504	000012	
12852	053314	005015	041523	050117	OPR019: .ASCII <15><12>@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@
12853	053322	035105	041440	030510	
12854	053330	024040	051124	043511	
12855	053336	026051	042440	031465	
12856	053344	034055	020073	044103	
12857	053352	026062	042440	034464	
12858	053360	031055	024040	041501	
12859	053366	041440	052517	046120	
12860	053374	026105	027040	053062	
12861	053402	041457	024515		
12862	053406	005015	042101	052512	.ASCIZ <15><12>/ADJUST R72 FOR CONSTANT LEVEL ON CH2/<15><12>
12863	053414	052123	051040	031067	
12864	053422	043040	051117	041440	
12865	053430	047117	052123	047101	
12866	053436	020124	042514	042526	
12867	053444	020114	047117	041440	
12868	053452	031110	005015	000	
12869	053457	015	050012	047522	OPR020: .ASCIZ <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12>
12870	053464	051107	046501	044040	
12871	053472	046101	020124	042520	
12872	053500	042116	047111	020107	
12873	053506	020055	040503	052122	
12874	053514	044522	043504	020105	
12875	053522	047506	046522	052101	
12876	053530	041040	044505	043516	
12877	053536	041440	051117	042522	
12878	053544	052103	042105	005015	
12879	053552	000			
12880	053553	015	025012	025052	OPR021: .ASCIZ <15><12>/***** PROGRAM HALTED *****/<15><12>
12881	053560	025052	020040	051120	
12882	053566	043517	040522	020115	

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35MACY11 30A(1052) 10-JAN-78 09:57 PAGE 238
ASCII MESSAGES

SEQ 0238

12883	053574	040510	052114	042105	
12884	053602	020040	025052	025052	
12885	053610	006452	000012		
12886	053614	040503	052122	044522	OPR022: .ASCIZ /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>
12887	053622	043504	020105	047506	
12888	053630	046522	052101	041440	
12889	053636	051117	042522	052103	
12890	053644	047511	020116	040506	
12891	053652	046111	042105	005015	
12892	053660	000			
12893	053661	015	050012	047522	ABORT: .ASCIZ <15><12>/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>
12894	053666	051107	046501	040440	
12895	053674	047502	052122	047111	
12896	053702	020107	042502	040503	
12897	053710	051525	020105	051105	
12898	053716	047522	020122	044124	
12899	053724	042522	044123	046117	
12900	053732	020104	054105	042503	
12901	053740	042105	042105	005015	
12902	053746	000			
12903					
12904					.SBTTL ERROR MESSAGES
12905	053747	106	052101	046101	EM1: .ASCIZ /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/
12906	053754	047055	047117	042440	
12907	053762	044530	052123	047101	
12908	053770	020124	042515	047515	
12909	053776	054522	040440	020124	
12910	054004	045522	030466	020061	
12911	054012	040502	042523	040440	
12912	054020	042104	042522	051523	
12913	054026	000			
12914	054027	106	052101	046101	EM2: .ASCIZ /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/
12915	054034	053455	044522	042524	
12916	054042	051040	040505	054504	
12917	054050	040440	042116	044440	
12918	054056	020105	044504	020104	
12919	054064	047516	020124	040503	
12920	054072	051525	020105	047111	
12921	054100	042524	051122	050125	
12922	054106	000124			
12923	054110	040506	040524	026514	EM3: .ASCIZ /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
12924	054116	040520	044522	054524	
12925	054124	042440	051122	051117	
12926	054132	052040	040522	027120	
12927	054140	050040	020103	052101	
12928	054146	042440	051122	051117	
12929	054154	036440	000040		
12930	054160	054105	042520	052103	EM4: .ASCIZ /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS: /
12931	054166	042105	044440	052116	
12932	054174	051105	052522	052120	
12933	054202	042040	042111	047040	
12934	054210	052117	047440	041503	
12935	054216	051125	047440	020122	
12936	054224	040527	020123	040514	
12937	054232	042524	020056	047503	
12938	054240	046515	047101	020104	

12939	054246	040527	035123	000		
12940	054253	123	041125	054523	EM5:	.ASCIZ /SUBSYSTEM CLEAR DID NOT RESET ERROR/
12941	054260	052123	046505	041440		
12942	054266	042514	051101	042040		
12943	054274	042111	047040	052117		
12944	054302	051040	051505	052105		
12945	054310	042440	051122	051117		
12946	054316	000				
12947	054317	123	041125	054523	EM6:	.ASCIZ /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
12948	054324	052123	046505	041440		
12949	054332	042514	051101	042040		
12950	054340	042111	047040	052117		
12951	054346	051040	051505	052105		
12952	054354	042040	053105	041511		
12953	054362	020105	047111	042524		
12954	054370	051122	050125	000124		
12955	054376	052523	051502	051531	EM7:	.ASCIZ /SUBSYSTEM CLEAR DID NOT SET READY/
12956	054404	042524	020115	046103		
12957	054412	040505	020122	044504		
12958	054420	020104	047516	020124		
12959	054426	042523	020124	042522		
12960	054434	042101	000131			
12961	054440	047527	042122	041440	EM10:	.ASCIZ /WORD COUNT INCORRECT/
12962	054446	052517	052116	044440		
12963	054454	041516	051117	042522		
12964	054462	052103	000			
12965	054465	102	051525	040440	EM11:	.ASCIZ /BUS ADDRESS INCORRECT/
12966	054472	042104	042522	051523		
12967	054500	044440	041516	051117		
12968	054506	042522	052103	000		
12969	054513	125	050120	051105	EM11A:	.ASCIZ /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
12970	054520	041040	051525	040440		
12971	054526	042104	042522	051523		
12972	054534	041040	052111	020123		
12973	054542	047111	047503	051122		
12974	054550	041505	020124	041050		
12975	054556	030501	026066	030440		
12976	054564	024467	000			
12977	054567	103	046131	047111	EM12:	.ASCIZ /CYLINDER ADDRESS INCORRECT/
12978	054574	042504	020122	042101		
12979	054602	051104	051505	020123		
12980	054610	047111	047503	051122		
12981	054616	041505	000124			
12982	054622	051124	041501	020113	EM13:	.ASCIZ /TRACK ADDRESS INCORRECT/
12983	054630	042101	051104	051505		
12984	054636	020123	047111	047503		
12985	054644	051122	041505	000124		
12986	054652	042523	052103	051117	EM14:	.ASCIZ /SECTOR ADDRESS INCORRECT./
12987	054660	040440	042104	042522		
12988	054666	051523	044440	041516		
12989	054674	051117	042522	052103		
12990	054702	000056				
12991	054704	047516	026516	054105	EM15:	.ASCIZ /NON-EXISTANT DRIVE/
12992	054712	051511	040524	052116		
12993	054720	042040	044522	042526		
12994	054726	000				

H03

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 240
ERROR MESSAGES

SEQ 0240

12995	054727	103	047117	051124	EM16:	.ASCIZ	/CONTROLLER TIMEOUT/
12996	054734	046117	042514	020122			
12997	054742	044524	042515	052517			
12998	054750	000124					
12999	054752	047125	052111	043040	EM17:	.ASCIZ	/UNIT FIELD ERROR/
13000	054760	042511	042114	042440			
13001	054766	051122	051117	000			
13002	054773	115	046125	050111	EM18:	.ASCIZ	/MULIPLE DRIVE SELECT/
13003	055000	042514	042040	044522			
13004	055006	042526	051440	046105			
13005	055014	041505	000124				
13006	055020	051120	043517	040522	EM19:	.ASCIZ	/PROGRAMMING ERROR/
13007	055026	046515	047111	020107			
13008	055034	051105	047522	000122			
13009	055042	047516	026516	054105	EM20:	.ASCIZ	/NON-EXISTANT MEMORY/
13010	055050	051511	040524	052116			
13011	055056	046440	046505	051117			
13012	055064	000131					
13013	055066	047125	041111	051525	EM21:	.ASCIZ	/UNIBUS PARITY ERROR/
13014	055074	050040	051101	052111			
13015	055102	020131	051105	047522			
13016	055110	000122					
13017	055112	046111	042514	040507	EM22:	.ASCIZ	/ILLEGAL FUNCTION/
13018	055120	020114	052506	041516			
13019	055126	044524	047117	000			
13020	055133	105	041503	044040	EM23:	.ASCIZ	/ECC HARD/
13021	055140	051101	000104				
13022	055144	040504	040524	041440	EM24:	.ASCIZ	/DATA CHECK/
13023	055152	042510	045503	000			
13024	055157	127	044522	042524	EM25:	.ASCIZ	/WRITE CHECK/
13025	055164	041440	042510	045503			
13026	055172	000					
13027	055173	104	052101	020101	EM26:	.ASCIZ	/DATA LATE/
13028	055200	040514	042524	000			
13029	055205	117	042520	040522	EM27:	.ASCIZ	/OPERATION INCOMPLETE/
13030	055212	044524	047117	044440			
13031	055220	041516	046517	046120			
13032	055226	052105	000105				
13033	055232	042510	042101	051105	EM28:	.ASCIZ	/HEADER VRC/
13034	055240	053040	041522	000			
13035	055245	102	042101	051440	EM29:	.ASCIZ	/BAD SECTOR ERROR/
13036	055252	041505	047524	020122			
13037	055260	051105	047522	000122			
13038	055266	047503	052116	047522	EM30:	.ASCIZ	/CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
13039	055274	046114	051105	042040			
13040	055302	052105	041505	042524			
13041	055310	020104	051104	053111			
13042	055316	020105	052502	020123			
13043	055324	040520	044522	054524			
13044	055332	042440	051122	051117			
13045	055340	000					
13046	055341	123	042505	020113	EM31:	.ASCIZ	/SEEK INCOMPLETE/
13047	055346	047111	047503	050115			
13048	055354	042514	042524	000			
13049	055361	116	047117	042455	EM32:	.ASCIZ	/NON-EXECUTABLE DRIVE FUNCTION/
13050	055366	042530	052503	040524			

13051	055374	046102	020105	051104	
13052	055402	053111	020105	052506	
13053	055410	041516	044524	047117	
13054	055416	000			
13055	055417	104	044522	042526	EM33: .ASCIZ /DRIVE DETECTED DRIVE BUS PARITY ERROR/
13056	055424	042040	052105	041505	
13057	055432	042524	020104	051104	
13058	055440	053111	020105	052502	
13059	055446	020123	040520	044522	
13060	055454	054524	042440	051122	
13061	055462	051117	000		
13062	055465	106	051117	040515	EM34: .ASCIZ /FORMAT ERROR/
13063	055472	020124	051105	047522	
13064	055500	000122			
13065	055502	051104	053111	020105	EM35: .ASCIZ /DRIVE TYPE ERROR/
13066	055510	054524	042520	042440	
13067	055516	051122	051117	000	
13068	055523	101	020103	047514	EM36: .ASCIZ /AC LOW/
13069	055530	000127			
13070	055532	050123	047111	046104	EM37: .ASCIZ /SPINDLE SPEED LOSS/
13071	055540	020105	050123	042505	
13072	055546	020104	047514	051523	
13073	055554	000			
13074	055555	104	044522	042526	EM38: .ASCIZ /DRIVE OFF TRACK/
13075	055562	047440	043106	052040	
13076	055570	040522	045503	000	
13077	055575	103	046131	047111	EM39: .ASCIZ /CYLINDER OVERFLOW/
13078	055602	042504	020122	053117	
13079	055610	051105	046106	053517	
13080	055616	000			
13081	055617	111	046114	043505	EM40: .ASCIZ /ILLEGAL DISK PACK ADDRESS/
13082	055624	046101	042040	051511	
13083	055632	020113	040520	045503	
13084	055640	040440	042104	042522	
13085	055646	051523	000		
13086	055651	127	044522	042524	EM41: .ASCIZ /WRITE LOCK ERROR/
13087	055656	046040	041517	020113	
13088	055664	051105	047522	000122	
13089	055672	051104	053111	020105	EM42: .ASCIZ /DRIVE TIMING ERROR/
13090	055700	044524	044515	043516	
13091	055706	042440	051122	051117	
13092	055714	000			
13093	055715	116	020117	042503	EM43: .ASCIZ /NO CERR WITH SOME OTHER ERROR SET/
13094	055722	051122	053440	052111	
13095	055730	020110	047523	042515	
13096	055736	047440	044124	051105	
13097	055744	042440	051122	051117	
13098	055752	051440	052105	000	
13099	055757	104	044522	042526	EM44: .ASCIZ /DRIVE UNSAFE/
13100	055764	052440	051516	043101	
13101	055772	000105			
13102	055774	042503	051122	051440	EM45: .ASCIZ /CERR SET BUT NO OTHER ERROR SET/
13103	056002	052105	041040	052125	
13104	056010	047040	020117	052117	
13105	056016	042510	020122	051105	
13106	056024	047522	020122	042523	

13107	056032	000124				
13108						
13109	056034	053126	042040	042111	EM46:	.ASCIZ /VV DID NOT SET WITH PACK ACK/
13110	056042	047040	052117	051440		
13111	056050	052105	053440	052111		
13112	056056	020110	040520	045503		
13113	056064	040440	045503	000		
13114	056071	123	040524	052524	EM47:	.ASCIZ /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
13115	056076	020123	040526	044514		
13116	056104	020104	042523	020124		
13117	056112	047117	051440	046105		
13118	056120	041505	020124	047524		
13119	056126	047040	047117	042455		
13120	056134	044530	052123	047101		
13121	056142	020124	051104	053111		
13122	056150	000105				
13123	056152	052123	052101	051525	EM48:	.ASCIZ /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
13124	056160	053040	046101	042111		
13125	056166	051040	051505	052105		
13126	056174	047440	020116	042523		
13127	056202	042514	052103	052040		
13128	056210	020117	054105	051511		
13129	056216	040524	052116	042040		
13130	056224	044522	042526	000		
13131	056231	123	040524	052524	EM49:	.ASCIZ /STATUS VALID NOT RESET ON DRIVE RELEASE/
13132	056236	020123	040526	044514		
13133	056244	020104	047516	020124		
13134	056252	042522	042523	020124		
13135	056260	047117	042040	044522		
13136	056266	042526	051040	046105		
13137	056274	040505	042523	000		
13138	056301	105	050130	041505	EM50:	.ASCIZ /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
13139	056306	042524	020104	047062		
13140	056314	020104	047111	042524		
13141	056322	051122	050125	020124		
13142	056330	044504	020104	047516		
13143	056336	020124	041517	052503		
13144	056344	020122	051117	053440		
13145	056352	051501	046040	052101		
13146	056360	027105	041440	046517		
13147	056366	040515	042116	053440		
13148	056374	051501	000072			
13149	056400	040503	047116	052117	EM51:	.ASCII /CANNOT READ BAD SECTOR FILE/<<15><12>
13150	056406	051040	040505	020104		
13151	056414	040502	020104	042523		
13152	056422	052103	051117	043040		
13153	056430	046111	006505	012		
13154	056435	124	051505	044524		.ASCIZ /TESTING ABORTED ON THIS DRIVE/
13155	056442	043516	040440	047502		
13156	056450	052122	042105	047440		
13157	056456	020116	044124	051511		
13158	056464	042040	044522	042526		
13159	056472	000				
13160	056473	101	044514	047107	EM52:	.ASCII /ALIGNMENT PACK ON THIS DRIVE./<<15><12>
13161	056500	042515	052116	050040		
13162	056506	041501	020113	047117		

K03

CZR6KE0 RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 243
ERROR MESSAGES

SEQ 0243

13163	056514	052040	044510	020123	
13164	056522	051104	053111	027105	
13165	056530	005015			
13166	056532	042524	052123	047111	.ASCIZ /TESTING ABORTED ON THIS DRIVE./
13167	056540	020107	041101	051117	
13168	056546	042524	020104	047117	
13169	056554	052040	044510	020123	
13170	056562	051104	053111	027105	
13171	056570	000			
13172	056571	124	020117	040515	EM53: .ASCIZ /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
13173	056576	054516	041040	042101	
13174	056604	051440	041505	047524	
13175	056612	051522	020056	042524	
13176	056620	052123	047111	020107	
13177	056626	041101	051117	042524	
13178	056634	020104	047117	052040	
13179	056642	044510	020123	051104	
13180	056650	053111	000105		
13181	056654	042510	042101	051105	EM54: .ASCIZ /HEADER WORD MISCOMPARE/
13182	056662	053440	051117	020104	
13183	056670	044515	041523	046517	
13184	056676	040520	042522	000	
13185	056703	104	052101	020101	EM55: .ASCIZ /DATA WORD MISCOMPARE/
13186	056710	047527	042122	046440	
13187	056716	051511	047503	050115	
13188	056724	051101	000105		
13189					
13190	056730	040503	047116	052117	EM56: .ASCIZ /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
13191	056736	043040	047111	020104	
13192	056744	042510	042101	051105	
13193	056752	030040	044440	020116	
13194	056760	042522	042101	040440	
13195	056766	042116	051440	051117	
13196	056774	020124	042510	042101	
13197	057002	051105	000123		
13198	057006	052123	052101	051525	EM5VAL: .ASCIZ /STATUS VALID/
13199	057014	053040	046101	042111	
13200	057022	000			
13201	057023	123	046105	041505	EMNZPR: .ASCIZ /SELECT WITH NON-ZERO STATUS PAIR/
13202	057030	020124	044527	044124	
13203	057036	047040	047117	055055	
13204	057044	051105	020117	052123	
13205	057052	052101	051525	050040	
13206	057060	044501	000122		
13207	057064	051127	052111	047111	EMWCS2: .ASCIZ /WRITING COMMAND STATUS REGISTER 2/
13208	057072	020107	047503	046515	
13209	057100	047101	020104	052123	
13210	057106	052101	051525	051040	
13211	057114	043505	051511	042524	
13212	057122	020122	000062		
13213		055502			EMDTPE= EM35 ;DRIVE TYPE ERROR
13214	057126	051104	053111	020105	EMDI: .ASCIZ /DRIVE INTERRUPT/
13215	057134	047111	042524	051122	
13216	057142	050125	000124		
13217		055417			EMDPAR= EM33 ;DRIVE DETECTED DRIVE BUS PARITY ERROR
13218	057146	051104	053111	020105	EMDSC: .ASCIZ /DRIVE STATUS CHANGE/

13219	057154	052123	052101	051525	
13220	057162	041440	040510	043516	
13221	057170	000105			
13222	057172	051104	053111	020105	EMDA: .ASCIZ /DRIVE ATTENTION/
13223	057200	052101	042524	052116	
13224	057206	047511	000116		
13225	057212	047503	052116	047522	EMCCLR: .ASCIZ /CONTROLLER CLEAR/
13226	057220	046114	051105	041440	
13227	057226	042514	051101	000	
13228		050536			EMSELD= OPER00 ;DRIVE SELECT
13229		050564			EMDCLR= OPER04 ;DRIVE CLEAR
13230		050625			EMRCAL= OPER12 ;RECALIBRATE
13231	057233	123	041505	047117	EM2INT: .ASCIZ /SECOND INTERRUPT/
13232	057240	020104	047111	042524	
13233	057246	051122	050125	000124	
13234	057254	042523	045505	052040	EMSKSF: .ASCIZ /SEEK TO SELF/
13235	057262	020117	042523	043114	
13236	057270	000			
13237		050650			EMSK= OPER16 ;SEEK
13238	057271	125	042516	050130	EMUXIT: .ASCIZ /UNEXPECTED INTERRUPT/
13239	057276	041505	042524	020104	
13240	057304	047111	042524	051122	
13241	057312	050125	000124		
13242	057316	047125	041111	051525	EMUR: .ASCIZ /UNIBUS RESET/
13243	057324	051040	051505	052105	
13244	057332	000			
13245	057333	122	051505	052105	EMRSET: .ASCIZ /RESET/
13246	057340	000			
13247		055173			EMDLT= EM26 ;DATA LATE
13248	057341	122	040505	044504	EMRDB: .ASCIZ /READING DATA BUFFER/
13249	057346	043516	042040	052101	
13250	057354	020101	052502	043106	
13251	057362	051105	000		
13252	057365	103	047117	051124	EMCERR: .ASCIZ /CONTROLLER ERROR/
13253	057372	046117	042514	020122	
13254	057400	051105	047522	000122	
13255	057406	052523	051502	051531	EMSCLR: .ASCIZ /SUBSYSTEM CLEAR/
13256	057414	042524	020115	046103	
13257	057422	040505	000122		
13258	057426	052515	052114	050111	EMMI: .ASCIZ /MULTIPLE INTERRUPTS/
13259	057434	042514	044440	052116	
13260	057442	051105	052522	052120	
13261	057450	000123			
13262	057452	040502	020104	042523	DRVABT: .ASCII /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))<<15><12>
13263	057460	052103	051117	020123	
13264	057466	047117	050040	041501	
13265	057474	020113	047111	040440	
13266	057502	042522	051501	052440	
13267	057510	042523	020104	054502	
13268	057516	052040	051505	020124	
13269	057524	041450	046131	031440	
13270	057532	031061	034050	024451	
13271	057540	005015			
13272	057542	042524	052123	047111	.ASCIZ /TESTING ABORTED ON THIS DRIVE/
13273	057550	020107	041101	051117	
13274	057556	042524	020104	047117	

M03

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 245
ERROR MESSAGES

SEQ 0245

13275	057564	052040	044510	020123
13276	057572	051104	053111	000105
13277				
13278				
13279	057600	051524	020124	052516
13280	057606	020115	051105	020122
13281	057614	041520	020040	051104
13282	057622	053111	000105	
13283	057626	051524	020124	052516
13284	057634	020115	051105	020122
13285	057642	041520	020040	051104
13286	057650	053111	000105	
13287	057654	045522	051503	020061
13288	057662	020040	045522	051503
13289	057670	020062	020040	045522
13290	057676	051504	020040	020040
13291	057704	045522	051105	020040
13292	057712	020040	045522	051501
13293	057720	043117	020040	045522
13294	057726	041504	046131	020040
13295	057734	045522	040504	000
13296	057741	122	041113	020101
13297	057746	020040	051040	053513
13298	057754	000103		
13299	057756	030101	020060	020040
13300	057764	020040	030102	020060
13301	057772	020040	020040	030101
13302	060000	020061	020040	020040
13303	060006	030102	020061	020040
13304	060014	020040	030101	020062
13305	060022	020040	020040	030102
13306	060030	020062	020040	020040
13307	060036	030101	020063	020040
13308	060044	020040	030102	000063
13309	060052	045522	051503	020061
13310	060060	020040	045522	051503
13311	060066	020062	020040	045522
13312	060074	051504	020040	020040
13313	060102	045522	051105	020040
13314	060110	020040	045522	051501
13315	060116	043117	020040	045522
13316	060124	051115	000061	
13317	060130	044124	020105	041101
13318	060136	053117	020105	051101
13319	060144	020105	054105	042520
13320	060152	052103	042105	042440
13321	060160	051122	051117	020123
13322	060166	044124	052101	042040
13323	060174	042111	047040	052117
13324	060202	051440	052105	044440
13325	060210	020116	050117	051105
13326	060216	052101	047511	035116
13327	060224	000		
13328	060225	124	042510	040440
13329	060232	047502	042526	040440
13330	060240	042522	052440	042516

.SBTTL DATA HEADERS FOR ERROR REPORTS
DH001: .ASCIZ /TST NUM ERR PC DRIVE/

DH002A: .ASCIZ /TST NUM ERR PC DRIVE/

DH002B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA/

DH002C: .ASCIZ /RKBA RKWC/

DH002D: .ASCIZ /A00 B00 A01 B01 A02 B02 A03 B03/

DH003B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKMR1/

DH005: .ASCIZ /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/

DH006: .ASCIZ /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/

13331	060246	050130	041505	042524
13332	060254	020104	051105	047522
13333	060262	051522	051440	052105
13334	060270	044440	020116	050117
13335	060276	051105	052101	047511
13336	060304	035116	000	
13337	060307	124	042510	040440
13338	060314	047502	042526	040440
13339	060322	042522	042440	051122
13340	060330	051117	020123	042523
13341	060336	020124	047111	047440
13342	060344	042520	040522	044524
13343	060352	047117	000072	
13344	060356	047101	020131	044506
13345	060364	046105	020104	044527
13346	060372	044124	040440	046114
13347	060400	047440	042516	020123
13348	060406	052515	052123	041040
13349	060414	020105	047503	051516
13350	060422	042111	051105	042105
13351	060430	044440	053116	046101
13352	060436	042111	000	
13353	060441	105	051122	051117
13354	060446	040440	020124	047503
13355	060454	050115	042514	044524
13356	060462	047117	047440	020106
13357	060470	050117	051105	052101
13358	060476	047511	000116	
13359	060502	054105	052120	020104
13360	060510	020040	051511	000
13361	060515	122	040505	020104
13362	060522	040504	040524	053440
13363	060530	052111	020110	047506
13364	060536	041522	042105	041440
13365	060544	046131	047111	042504
13366	060552	020122	053117	051105
13367	060560	046106	053517	000
13368	060565	116	052117	051440
13369	060572	052105	040440	020123
13370	060600	020101	042522	052523
13371	060606	052114	047440	000106
13372	060614	047516	020124	042522
13373	060622	042523	020124	051501
13374	060630	040440	051040	051505
13375	060636	046125	020124	043117
13376	060644	000		
13377	060645	123	052105	040440
13378	060652	020123	020101	042522
13379	060660	052523	052114	047440
13380	060666	000106		
13381	060670	042522	042523	020124
13382	060676	051501	040440	051040
13383	060704	051505	046125	020124
13384	060712	043117	000	
13385	060715	107	047517	020104
13386	060722	020040	041040	042101

DH007: .ASCIZ /THE ABOVE ARE ERRORS SET IN OPERATION:/

DH005A: .ASCIZ /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/

DH010: .ASCIZ /ERROR AT COMPLETION OF OPERATION/

DH010A: .ASCIZ /EXPTD IS/

DH010B: .ASCIZ /READ DATA WITH FORCED CYLINDER OVERFLOW/

DH011: .ASCIZ /NOT SET AS A RESULT OF/

DH012: .ASCIZ /NOT RESET AS A RESULT OF/

DH013: .ASCIZ /SET AS A RESULT OF/

DH014: .ASCIZ /RESET AS A RESULT OF/

DH015: .ASCIZ /GOOD BAD WORD NUM/

13387	060730	020040	020040	053440	
13388	060736	051117	020104	052516	
13389	060744	000115			
13390	060746	052523	051502	051531	DH016: .ASCIZ /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
13391	060754	042524	020115	046103	
13392	060762	040505	020122	047524	
13393	060770	051040	051505	052105	
13394	060776	051440	042505	020113	
13395	061004	044514	044515	020124	
13396	061012	046101	047514	044527	
13397	061020	043516	044040	040505	
13398	061026	051504	052040	020117	
13399	061034	042522	047514	042101	
13400	061042	000			
13401	061043	123	046105	041505	DH017: .ASCIZ /SELECT AFTER RECAL/
13402	061050	020124	043101	042524	
13403	061056	020122	042522	040503	
13404	061064	000114			
13405	061066	042522	040503	020114	DH018: .ASCIZ /RECAL COMPLETE ATTN AFTER SELECT/
13406	061074	047503	050115	042514	
13407	061102	042524	040440	052124	
13408	061110	020116	043101	042524	
13409	061116	020122	042523	042514	
13410	061124	052103	000		
13411					
13412					.SBTTL DATA TABLES FOR ERROR REPORTS
13413		061130			.EVEN
13414	061130	001302	001116	001626	DT003: .WORD \$TESTN,\$ERRPC,DRVNUM,T.CS1,T.CS2,T.DS,T.ER,T.ASOF,T.MR1
13415	061136	001540	001550	001552	
13416	061144	001554	001556	001566	
13417	061152	001302	001116	001626	DT001: .WORD \$TESTN,\$ERRPC,DRVNUM
13418	061160				DT002:
13419	061160				DT004:
13420	061160				DT005:
13421	061160				DT006:
13422	061160				DT007:
13423	061160	001302	001116	001626	DT010: .WORD \$TESTN,\$ERRPC,DRVNUM
13424	061166	001540	001550	001552	.WORD T.CS1,T.CS2,T.DS,T.ER,T.ASOF
13425	061174	001554	001556		
13426	061200	001560	001546		.WORD T.DCYL,T.DA
13427	061204	001544	001542		.WORD T.BA,T.WC
13428	061210	001202	001204	001206	.WORD \$REG10,\$REG11,\$REG12,\$REG13,\$REG14,\$REG15,\$REG16,\$REG17
13429	061216	001210	001212	001214	
13430	061224	001216	001220		
13431	061230	001302	001116	001626	DT015: .WORD \$TESTN,\$ERRPC,DRVNUM
13432	061236	001202	001204	001206	DT015A: .WORD \$REG10,\$REG11,\$REG12
13433					.SBTTL DATA FORMATS FOR ERROR REPORTS
13434	061244	000001			DF001: .WORD 1
13435	061246	003	000		.BYTE 3,0
13436					
13437	061250	000004			DF002: .WORD 4
13438	061252	000	000		.BYTE 0,0
13439	061254	057626			.WORD DH002A
13440	061256	003	000		.BYTE 3,0
13441	061260	057654			.WORD DH002B
13442	061262	007	000		.BYTE 7,0

13443	061264	057741		.WORD	DH002C
13444	061266	002	000	.BYTE	2,0
13445					
13446					
13447	061270	000002		DF003: .WORD	2
13448	061272	003	000	.BYTE	3,0
13449	061274	060052		.WORD	DH003B
13450	061276	006	000	.BYTE	6,0
13451					
13452	061300	000006		DF004: .WORD	6
13453	061302	000	000	.BYTE	0,0
13454	061304	000000		DF004A: .WORD	0
13455	061306	000	000	.BYTE	0,0
13456	061310	057626		.WORD	DH002A
13457	061312	003	000	.BYTE	3,0
13458	061314	057654		.WORD	DH002B
13459	061316	007	000	.BYTE	7,0
13460	061320	057741		.WORD	DH002C
13461	061322	002	000	.BYTE	2,0
13462	061324	057756		.WORD	DH002D
13463	061326	010	000	.BYTE	10,0
13464					
13465	061330	000007		DF005: .WORD	7
13466	061332	000	000	.BYTE	0,0
13467	061334	000000		DF005A: .WORD	0
13468	061336	000	000	.BYTE	0,0
13469	061340	057626		.WORD	DH002A
13470	061342	003	000	.BYTE	3,0
13471	061344	057654		.WORD	DH002B
13472	061346	007	000	.BYTE	7,0
13473	061350	057741		.WORD	DH002C
13474	061352	002	000	.BYTE	2,0
13475	061354	057756		.WORD	DH002D
13476	061356	010	000	.BYTE	10,0
13477	061360	060356		.WORD	DH005A
13478	061362	000	000	.BYTE	0,0
13479					
13480	061364	000005		DF006: .WORD	5
13481	061366	000	000	.BYTE	0,0
13482	061370	000000		DF006A: .WORD	0
13483	061372	000	000	.BYTE	0,0
13484	061374	057626		.WORD	DH002A
13485	061376	003	000	.BYTE	3,0
13486	061400	057654		.WORD	DH002B
13487	061402	007	000	.BYTE	7,0
13488	061404	057741		.WORD	DH002C
13489	061406	002	000	.BYTE	2,0
13490					
13491	061410	000004		DF007: .WORD	4
13492	061412	000	000	.BYTE	0,0
13493	061414	000000		DF007A: .WORD	0
13494	061416	000	000	.BYTE	0,0
13495	061420	057626		.WORD	DH002A
13496	061422	003	000	.BYTE	3,0
13497	061424	060052		.WORD	DH003B
13498	061426	006	000	.BYTE	6,0

```

13499
13500 061430 000004
13501 061432 000 000
13502 061434 000000
13503 061436 000 000
13504 061440 057626
13505 061442 003 000
13506 061444 060502
13507 061446 002 000
13508
13509 061450 000004
13510 061452 000 000
13511 061454 000000
13512 061456 000 000
13513 061460 057626
13514 061462 003 000
13515 061464 060052
13516 061466 000006 000000
13517
13518 061472 000002
13519 061474 003 000
13520 061476 060715
13521 061500 003 000
13522
13523 061502 000001
13524 061504 003 000
13525
13526 061506 000052
13527 061632 000052
13528 070376
13529 070376 000240
13530 070400 005014
13531 070402 005237 001662
13532 070406 000240
13533 070410 000002
13534 070412 000240
13535 070414 001000
13536 072414 001000
13537 000001

```

```

DF010: .WORD 4
        .BYTE 0,0
DF010A: .WORD 0
        .BYTE 0,0
        .WORD 04002A
        .BYTE 3,0
        .WORD 04010A
        .BYTE 2,0

DF011: .WORD 4
        .BYTE 0,0
DF011A: .WORD 0
        .BYTE 0,0
        .WORD 04002A
        .BYTE 3,0
        .WORD 04003B
        .WORD 6,0

DF015: .WORD 2
        .BYTE 3,0
        .WORD 04015
        .BYTE 3,0

DF016: .WORD 1
        .BYTE 3,0

BS26: .BLKW 52
BS24: .BLKW 52
      =70376
NOP
SPCHLR: CLR (R4)
        INC INTSET
        NOP
SPCPAR: RTI
        NOP
IBUFF: .BLKW 1000
OBUFF: .BLKW 1000
.END

```

```

;CLEAR MEM MAINT REG
;COUNT THE INTERRUPT
;RETURN

```

ABASE = 177440	1968#	2290	2331		
ABORT = 053661	9685	12893#			
ABTFAI = 044546	9480	11504#			
ACDW1 = 000000	2290	2333			
ACDW2 = 000000	2290				
ACLO = 000010	2063#	10598			
ACLOER = 000100	10600	12469#			
ACPUOP = 000000	2290	2305			
ADDW0 = 000000	2290				
ADDW1 = 000000	2290				
ADDW10 = 000000	2290				
ADDW11 = 000000	2290				
ADDW12 = 000000	2290				
ADDW13 = 000000	2290				
ADDW14 = 000000	2290				
ADDW15 = 000000	2290				
ADDW2 = 000000	2290				
ADDW3 = 000000	2290				
ADDW4 = 000000	2290				
ADDW5 = 000000	2290				
ADDW6 = 000000	2290				
ADDW7 = 000000	2290				
ADDW8 = 000000	2290				
ADDW9 = 000000	2290				
ADEVCT = 000000	2290	2296			
ADEVN = 000000	2290	2332			
ADJCLK = 044172	3264	11397#			
RENV = 000000	2290	2301			
REVM = 000000	2290	2302			
AFATAL = 000000	2290	2293			
AMADR1 = 000000	2290	2318			
AMADR2 = 000000	2290	2322			
AMADR3 = 000000	2290	2325			
AMADR4 = 000000	2290	2328			
AMAMS1 = 000000	2290	2312			
AMAMS2 = 000000	2290	2320			
AMAMS3 = 000000	2290	2323			
AMAMS4 = 000000	2290	2326			
AMSGAO = 000000	2290	2298			
AMSLG = 000000	2290	2299			
AMSGTY = 000000	2290	2292			
AMTYP1 = 000000	2290	2313			
AMTYP2 = 000000	2290	2321			
AMTYP3 = 000000	2290	2324			
AMTYP4 = 000000	2290	2327			
AFASS = 000000	2290	2295			
APRIOR = 000240	1967#	2290			
APTC SU = 000040	9538#	11537			
APTENV = 000001	9242	9494	9536#	9593	11530
APTSIZ = 000200	2804	9535#			
APTSPO = 000100	9496	9537#	11532		
ASWREG = 000000	2290	2303			
ATESTN = 000000	2290	2294			
AUNIT = 000000	2290	2297			
AUSWR = 000000	2290	2304			
AVECT1 = 000210	1966#	2290	2329		

E05

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 (9:35)

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 264
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0263

S.HDFL = 000200
S.HDMM = 000040
S.ICYL = 000040
S.ILF = 000400
S.LIMD = 020000
S.LOAD = 010000
S.MHD = 000400
S.NMOV = 010000
S.OFF = 002000
S.PAR = 001000
S.PIP = 020000
S.PLO = 004000
S.REV = 004000
S.RTZ = 020000
S.SECT = 000020
S.SKI = 002000
S.SPIN = 010000
S.SPLS = 010000
S.SPOK = 001000
S.TYPE = 000400
S.UNLD = 040000
S.UNS = 040000
S.VV = 000100
S.WCLK = 000040
S.WGAT = 000100
S.WLE = 004000
S.WRL = 004000
S.XDOK = 000020
S.XERR = 001000
TBITVE = 000014
TCHKOP = 104421

2136#
2120#
2106#
2109#
2142#
2127#
2137#
2141#
2098#
2110#
2101#
2140#
2126#
2128#
2133#
2111#
2100#
2113#
2124#
2096#
2129#
2115#
2094#
2134#
2135#
2112#
2099#
2119#
2138#
1921#
3084
3690
4636
5097
5801
6258
6945
7551
7865
8383
8904
3203
4651
8213
3070
9956
2695#
1928#
3065
3561
3972
4632
5076
5555
6053
6637

8018 8581

3150	3220	3234	3287	3296	3326	3342	3377	3509	3572	3599	3642
3778	3829	4078	4244	4269	4295	4373	4406	4452	4486	4522	4575
4695	4769	4748	4778	4793	4844	4867	4930	4948	5004	5017	5080
5152	5168	5230	5245	5303	5317	5366	5386	5440	5457	5767	5786
5851	5874	5889	5937	5961	6022	6042	6057	6108	6127	6142	6201
6316	6372	6428	6474	6543	6563	6626	6641	6754	6805	6820	6881
7035	7065	7108	7123	7165	7180	7301	7316	7430	7447	7494	7525
7570	7596	7621	7711	7729	7735	7750	7775	7790	7795	7810	7833
7871	7886	7961	8015	8104	8148	8239	8272	8298	8344	8360	8368
8457	8464	8479	8494	8537	8578	8613	8631	8682	8753	8822	8890
9056	9076	9104	11446	11453	11481	12176#					
3421	3465	3530	3659	3709	3937	3945	4006	4393	4474	4537	4590
5505	5558	5692	6502	6904	6980	7200	7338	7913	7991	8065	8140
8420	8554	12177#									
3120	3389	3606	3742	3757	3809	3850	3878	4273	8765	8832	8922
10176	10222	10322	10521	12175#							
9856#	9885#	9888#	9947#	10163#							
11752#	11753#										
3115	3197	3229	3283	3292	3323	3338	3373	3417	3461	3506	3526
3568	3595	3638	3648	3686	3696	3738	3774	3804	3846	3909	3928
4002	4075	4240	4265	4369	4386	4449	4467	4518	4533	4571	4586
4647	4691	4705	4744	4774	4789	4840	4863	4926	4945	5000	5014
5093	5148	5162	5226	5241	5299	5313	5362	5382	5436	5454	5502
5654	5763	5782	5797	5847	5870	5885	5933	5957	5971	6018	6038
6103	6123	6138	6197	6254	6312	6368	6424	6470	6498	6539	6622
6715	6801	6816	6877	6900	6941	6976	7031	7061	7104	7119	7161

TCHKWE = 104422
TGETRK = 104420
TIMCNT 001654
TKVEC = 000060
TLOADR = 104417

L05

CZR6KEO RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 271
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0270

\$SETUP= 000137	2745#	2772	2773	2775	2777	2779	2781	2782	2783	2785	2812	2815	9204
\$SIZE 032664	9341	9554	9603	9611	11773	11778	11779	11809	11985				
\$SIZEX 033126	2831	9265#											
\$STUP = 177777	9303	9313#											
\$SVLAD 033444	2745#												
\$SWR = 167400	9356	9395#											
	1797#	1809	1813	1814	1815	1816	1817	1818	1819	1820	2280	2281	2282
	2782	2783	2785	2786	2935	2966	3031	3276	3316	3366	3409	3453	3631
	3679	3731	3797	3839	3900	3965	4039	4218	4345	4421	4500	4552	4606
	4666	4725	4822	4897	4978	5046	5127	5197	5277	5346	5421	5490	5543
	5604	5747	5831	5917	6002	6087	6174	6233	6290	6347	6403	6457	6486
	6521	6601	6675	6779	6858	7012	7088	7144	7280	7410	7472	7677	7848
	7940	8046	8084	8164	8256	8312	8440	8647	8714	8784	8858	9024	9199
	9205	9232	9238	9240	9332	9333	9334	9335	9336	9342	9359	9361	9362
	9375	9376	9377	9384	9385	9386	9398	9401	9404	9545	9546	9547	9548
	9549	9558	9565	9600	9604	9616	12120						
\$SWREG 001320	2303#	2806											
\$SWRMK= 000000	1820	1821	9336	9337	9365								
\$SWOBT 033516	9371	9405#	9963	10225	10325	10952							
\$TERM = 000102	12192#												
\$TESTN 001302	2294#	9396*	9627*	9628*	9961	10223	10323	10950	11425*	13414	13417	13423	13431
\$TIMES 001262	2280#	2782*	2935*	2966*	3031*	3276*	3316*	3366*	3409*	3453*	3631*	3679*	3731*
	3797*	3839*	3900*	3965*	4039*	4218*	4345*	4421*	4500*	4552*	4606*	4666*	4725*
	4822*	4897*	4978*	5046*	5127*	5197*	5277*	5346*	5421*	5490*	5543*	5604*	5747*
	5831*	5917*	6002*	6087*	6174*	6233*	6290*	6347*	6403*	6457*	6486*	6521*	6601*
	6675*	6779*	6858*	7012*	7088*	7144*	7280*	7410*	7472*	7677*	7848*	7940*	8046*
	8084*	8164*	8256*	8312*	8440*	8647*	8687*	8714*	8784*	8858*	9024*	9205*	9384*
	9391	9394*	9404										
\$TKB 001146	2239#	11733	11754	11765	11790	11818	11845						
\$TKCNT 045514	11734#	11749*	11779	11796*	11910	11912*							
\$TKINT 045524	2763	3861	8758	11749#	11770	11831							
\$TKQEN= 045523	11738#	11804	11915										
\$TKQIN 045516	11735#	11750*	11751	11802*	11803*	11804	11806*						
\$TKQOU 045520	11736#	11751*	11913	11914*	11915	11917*							
\$TKQSR 045522	11737#	11750	11806	11917									
\$TKS 001144	2238#	5659*	5675*	5731*	6720*	6736*	6764*	11733	11755*	11786*	11788	11794*	11816
	11832*	11842	11854*	11874*									
\$TKSRV 045574	11752	11765#											
\$TMP0 001222	2264#	2869*	2871	2873	2880*	2882	2885	2897*	2899	2901	2903*	2904*	2905*
	2906#	2907*	2909	10311*	10315*								
\$TMP1 001224	2265#	8869*	8947*	8972	8983	9005*	9026*	9131*	9156	9167	9189*	10737	11118*
	11120#	11122	11136*	11152	11153	11157	11159*						
\$TMP10 001242	2272#	10511*	10691	10793	10923								
\$TMP11 001244	2273#	10512*	10772	10917									
\$TMP12 001246	2274#	10513*	10752	10912									
\$TMP13 001250	2275#	10515*	10519*	10549	10688	10706	10750	10770	10791	10842	10897		
\$TMP14 001252	2276#	10865*	10884	10888*	10890	10894*	10915	10919*	10921	10925*			
\$TMP15 001254	2277#	10866*	10871*										
\$TMP16 001256	2278#	10094*	10099*										
\$TMP17 001260	2279#	10096*	10104	10106*									
\$TMP2 001226	2266#	10738											
\$TMP3 001230	2267#	10250*	10264*	10279									
\$TMP4 001232	2268#	3062*	3091	3095*									
\$TMP5 001234	2269#	10975*	10985*	10992									
\$TMP6 001236	2270#	10972*	10993										
\$TMP7 001240	2271#	3183*	3253*	3258	10973*	10994							

COMMEN	1931#																	
ENDCOM	1931#																	
ERROR	1825#	2952	2981	3034	3068	3081	3085	3111	3118	3151	3190	3199	3207	3212	3221			
	3226	3232	3235	3240	3247	3278	3285	3288	3294	3297	3301	3318	3325	3327	3334			
	3340	3343	3350	3368	3375	3378	3385	3396	3411	3419	3425	3432	3455	3463	3469			
	3483	3488	3494	3501	3510	3516	3522	3528	3534	3539	3545	3551	3557	3570	3573			
	3579	3585	3597	3600	3611	3617	3633	3640	3643	3657	3663	3666	3681	3688	3691			
	3705	3713	3733	3740	3747	3755	3762	3769	3776	3780	3786	3799	3806	3815	3827			
	3830	3841	3848	3856	3876	3883	3904	3912	3917	3922	3930	3941	3949	3967	3975			
	3980	3985	4004	4010	4063	4077	4079	4090	4101	4111	4118	4127	4140	4203	4223			
	4239	4242	4245	4264	4267	4270	4278	4285	4293	4296	4297	4298	4304	4313	4348			
	4371	4374	4377	4388	4397	4407	4424	4451	4453	4457	4469	4478	4487	4503	4520			
	4523	4535	4541	4555	4573	4576	4588	4594	4609	4634	4637	4649	4655	4670	4693			
	4696	4707	4710	4728	4746	4749	4776	4779	4791	4794	4800	4808	4825	4842	4845			
	4865	4868	4874	4881	4912	4928	4931	4947	4949	4955	4962	4988	5002	5005	5016			
	5018	5024	5031	5062	5078	5081	5095	5098	5104	5112	5136	5150	5153	5164	5167			
	5173	5181	5213	5228	5231	5243	5246	5253	5261	5286	5301	5304	5315	5318	5324			
	5332	5348	5364	5367	5384	5387	5398	5406	5423	5438	5441	5456	5458	5469	5477			
	5492	5504	5509	5518	5524	5531	5545	5557	5562	5571	5578	5585	5606	5690	5696			
	5705	5712	5719	5725	5749	5765	5768	5784	5787	5799	5802	5808	5816	5833	5849			
	5852	5872	5875	5887	5890	5896	5904	5919	5935	5938	5959	5962	5973	5979	5987			
	6004	6020	6023	6040	6043	6055	6058	6064	6072	6089	6105	6109	6125	6128	6140			
	6143	6149	6157	6176	6199	6202	6208	6216	6235	6256	6259	6265	6273	6292	6314			
	6317	6323	6331	6349	6370	6373	6379	6387	6405	6426	6429	6435	6443	6459	6472			
	6475	6488	6500	6506	6527	6541	6544	6556	6564	6565	6566	6573	6581	6604	6624			
	6627	6639	6642	6648	6656	6700	6751	6755	6783	6803	6806	6818	6821	6827	6835			
	6861	6879	6882	6902	6908	6917	6926	6932	6943	6946	6952	6960	6978	6984	6990			
	6998	7018	7033	7036	7049	7063	7066	7090	7106	7109	7121	7124	7148	7163	7166			
	7178	7181	7187	7198	7204	7225	7234	7243	7251	7284	7299	7302	7314	7317	7324			
	7336	7342	7363	7372	7381	7389	7412	7428	7431	7445	7448	7492	7495	7512	7523			
	7526	7539	7547	7552	7568	7571	7573	7580	7593	7597	7608	7619	7622	7630	7638			
	7679	7698	7707	7712	7726	7730	7733	7736	7748	7751	7772	7776	7787	7791	7794			
	7796	7808	7811	7830	7834	7850	7863	7866	7870	7884	7887	7899	7911	7917				
	7919	7923	7943	7955	7959	7962	7974	7977	7989	7995	7998	8013	8016	8022	8033			
	8048	8063	8069	8086	8102	8105	8138	8144	8149	8167	8179	8182	8194	8212	8217			
	8221	8226	8238	8240	8258	8270	8273	8276	8296	8299	8314	8326	8330	8342	8345			
	8358	8361	8366	8369	8381	8384	8418	8424	8443	8455	8458	8462	8465	8477	8480			
	8492	8495	8507	8535	8538	8552	8558	8561	8576	8579	8585	8592	8597	8611	8614			
	8629	8632	8650	8661	8665	8680	8683	8719	8734	8750	8754	8772	8789	8803	8819			
	8823	8839	8880	8889	8891	8902	8905	8911	8920	8928	9036	9045	9049	9054	9057			
	9067	9070	9074	9077	9089	9102	9105	9112	11428	11444	11447	11451	11454	11457	11479			
ESCAPE	11482																	
GETPRI	1931#																	
GETSWR	1931#	9273																
INITSS	2741#	2815#																
	5748	4237	4262	4291	4824	4910	4986	5060	5134	5211	5284	5422	5491	5544	5605			
	6699	5832	5918	6003	6088	6175	6234	6291	6348	6404	6458	6487	6525	6554	6603			
	7678	6782	6860	6930	7016	7047	7089	7147	7186	7283	7323	7411	7510	7578	7606			
	8560	7849	7918	7942	7976	7997	8021	8047	8085	8166	8219	8257	8275	8313	8442			
LOADLS	2743#	8584	8591	8649	8663	8718	8788	8878	9034	11427	11456							
	5835	4827	4847	4938	4989	5086	5137	5233	5288	5425	5443	5616	5751	5774	5789			
	6360	5858	5877	5921	5949	5963	6006	6030	6045	6091	6115	6130	6189	6246	6304			
	7168	6416	6614	6629	6701	6793	6808	6863	6892	6933	6968	7050	7092	7111	7153			
	7874	7188	7289	7304	7326	7418	7435	7513	7558	7581	7683	7738	7753	7798	7813			
	8260	7889	7901	7945	7964	7979	8001	8050	8092	8111	8128	8169	8184	8196	8228			
		8278	8316	8332	8347	8371	8407	8445	8467	8482	8497	8509	8540	8563	8601			

MSG	8619	8670	11434	11463											
	2927#	2929	2958#	2960	2987#	2990	3268#	3270	3305#	3307	3357#	3359	3401#	3403	3434#
	3436	3620#	3622	3667#	3669	3717#	3719	3738#	3790	3831#	3833	3889#	3891	3955#	3957
	4014#	4016	4208#	4210	4328#	4330	4410#	4412	4490#	4492	4543#	4545	4595#	4597	4656#
	4658	4715#	4717	4812#	4814	4886#	4888	4967#	4969	5035#	5037	5116#	5118	5186#	5188
	5266#	5268	5337#	5339	5410#	5412	5482#	5484	5535#	5537	5589#	5591	5738#	5740	5821#
	5823	5908#	5910	5992#	5994	6077#	6079	6162#	6164	6221#	6223	6278#	6280	6335#	6337
	6391#	6393	6447#	6449	6476#	6478	6511#	6513	6592#	6594	6661#	6663	6769#	6771	6843#
	6845	7003#	7005	7079#	7081	7125#	7127	7261#	7263	7399#	7401	7452#	7454	7659#	7661
	7837#	7839	7929#	7932	8037#	8039	8071#	8074	8152#	8155	8244#	8246	8300#	8302	8425#
	8427	8634#	8636	8706#	8708	8775#	8777	8842#	8844	9008#	9010				
MULT	1931#														
NEWST	1931#	2927	2958	2988	3268	3305	3357	3401	3434	3620	3667	3717	3788	3831	3889
	3955	4014	4208	4328	4410	4490	4543	4595	4656	4715	4812	4886	4967	5035	5116
	5186	5266	5337	5410	5482	5535	5589	5738	5821	5908	5992	6077	6162	6221	6278
	6335	6391	6447	6476	6511	6592	6661	6769	6843	7003	7079	7125	7261	7399	7452
OPCHK	7659	7837	7930	8037	8072	8153	8244	8300	8425	8634	8706	8775	8842	9008	
	2745#	4844	4867	4930	4948	5004	5017	5080	5097	5152	5166	5230	5245	5303	5317
	5440	5457	5767	5786	5801	5851	5874	5889	5937	5961	6022	6042	6057	6108	6127
	6142	6201	6258	6316	6372	6428	6474	6543	6626	6641	6754	6805	6820	6881	6945
	7035	7065	7108	7123	7165	7180	7301	7316	7430	7447	7494	7525	7550	7570	7595
	7621	7710	7729	7735	7750	7774	7790	7795	7810	7832	7865	7871	7886	7961	8015
	8104	8147	8239	8272	8298	8344	8360	8368	8383	8457	8464	8479	8494	8537	8578
POP	8613	8631	8682	8752	8821	8890	8904	9055	9075	9104	11446	11453	11481		
PUSH	1931#	9528	9529	9995	10129	10327	11195	11214	11218	11382	11639	12028	12077	12109	12110
REPORT	1931#	9489	9491	9512	9984	10116	10301	11084	11095	11598	12002	12057	12090	12096	
RESDC	1931#														
	2742#	4875	4956	5025	5106	5175	5255	5326	5471	5810	5898	5981	6066	6151	6210
RKLOAD	6267	6325	6381	6437	6575	6650	6829	6954	6992	7632	7632	5898	5981	6066	6151
	2744#	4840	4863	4926	4945	5000	5014	5076	5093	5148	5162	5226	5241	5299	5313
	5436	5454	5502	5555	5763	5782	5797	5847	5870	5885	5933	5957	5971	6018	6038
	6053	6103	6123	6138	6197	6254	6312	6368	6424	6470	6498	6622	6637	6801	6816
	6877	6900	6941	6976	7031	7061	7104	7119	7161	7176	7196	7297	7312	7334	7426
	7443	7490	7521	7566	7617	7724	7746	7785	7806	7861	7882	7897	7909	7953	7972
	7987	8011	8100	8136	8177	8192	8236	8268	8324	8340	8356	8379	8416	8453	8475
	8490	8505	8550	8573	8609	8627	8678	8732	8801	8887	8900	9043	9052	9065	9072
SCOPE	9087	11442	11477												
	1826#	2934	2965	3030	3275	3315	3365	3408	3452	3630	3678	3730	3796	3838	3899
	3964	4038	4217	4344	4420	4499	4551	4605	4665	4724	4821	4896	4977	5045	5126
	5196	5276	5345	5420	5489	5542	5603	5746	5830	5916	6001	6086	6173	6232	6289
	6346	6402	6456	6485	6520	6600	6674	6778	6857	7011	7087	7143	7279	7409	7471
	7676	7847	7939	8045	8083	8163	8255	8311	8439	8646	8686	8713	8783	8857	9023
	9203														
SETPRI	1931#	11906													
SETTRA	12150#	12159	12160	12161	12162	12164	12166	12167	12168	12169	12170	12171	12172	12173	12174
	12175	12176	12177	12178	12179	12180	12181	12182	12183	12184	12185	12186	12187	12188	12189
	12190	12191													
SETUP	1931#	2765													
SKIP	1931#														
SLASH	1931#														
SPACE	1931#														
STARS	1931#														
	3305	2189	2191	2198	2211	2286	2289	2927	2933	2958	2964	2988	3029	3268	3274
	3795	3314	3357	3364	3401	3407	3434	3451	3620	3629	3667	3677	3717	3729	3788
	4490	3831	3837	3889	3898	3955	3963	4014	4037	4208	4216	4328	4343	4410	4419
	4976	4498	4543	4550	4595	4604	4656	4664	4715	4723	4812	4820	4886	4895	4967
		5035	5044	5116	5125	5186	5195	5266	5275	5337	5344	5410	5419	5482	5488

D06

CZR6KED RK6 FCTNL CTRL DIAG
CZR6KE.P11 10-JAN-78 09:35

MACY11 30A(1052) 10-JAN-78 09:57 PAGE 277
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0275

. ABS. 074414 000

ERRORS DETECTED: 0

DSKZ:CZR6KE,DSKZ:CZR6KE.SEQ/SOL/CRF/NL:TOC/DOC=DSKM:CZR6KE.P11

RUN-TIME: 45 43 4 SECONDS

RUN-TIME RATIO: 689/92=7.4

CORE USED: 29K (57 PAGES)

DOCUMENT PAGES: 275