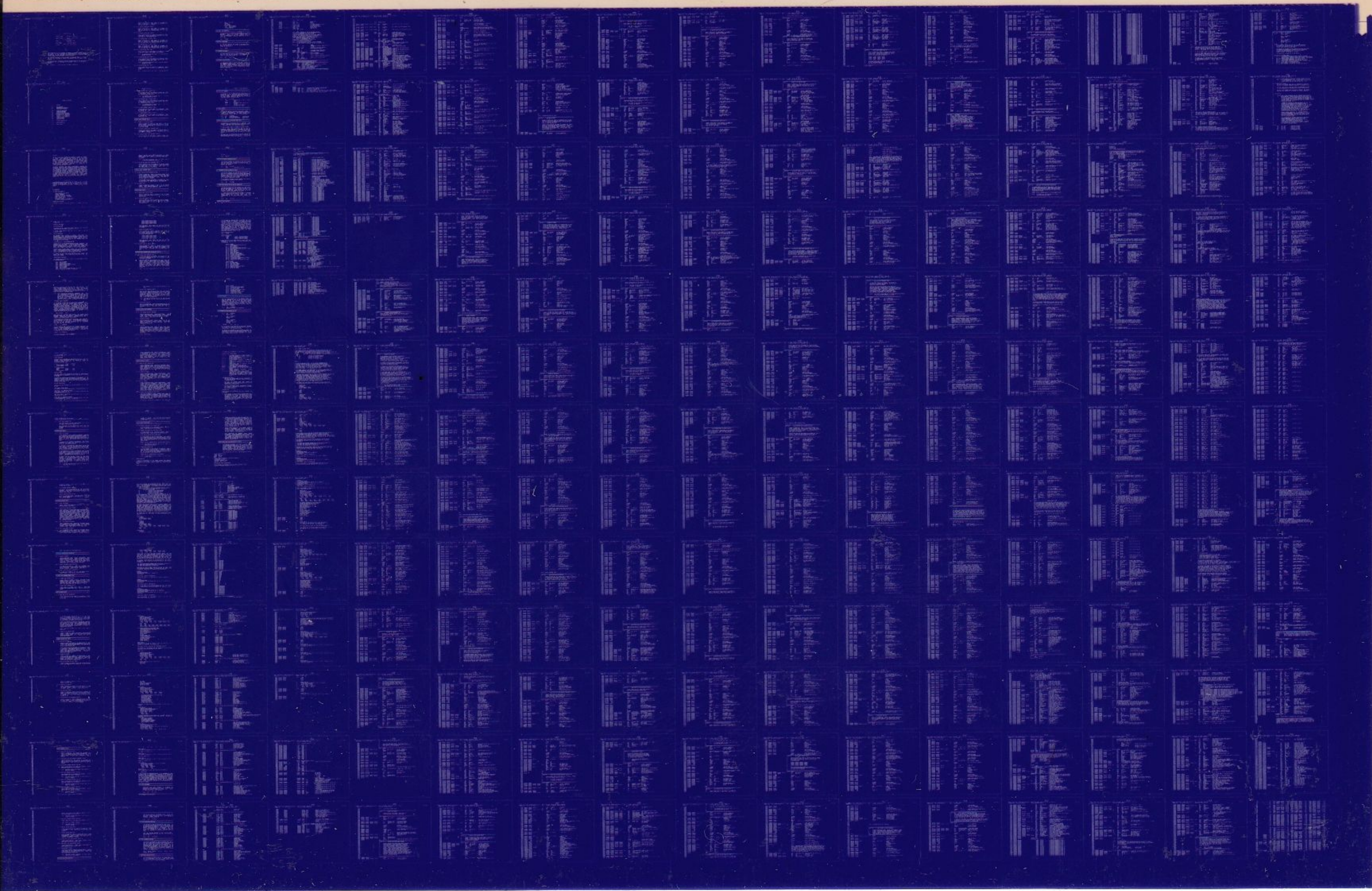


RK611

FUNCT. CONTROLLER DIAG.
MD-11-DZR6K-C

EP-DZR6K-C-DL-B
COPYRIGHT © 1976
FICHE 1 OF 2

DEC 1976
digital
MADE IN USA



RK611

FUNCT. CONTROLLER DIAG.
MD-11-DZR6K-C

EP-DZR6K-C-DL-B
COPYRIGHT © 1976
FICHE 2 OF 2

DEC 1976
digital
MADE IN USA



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 (DZR6K) IS A SERIES OF TESTS THAT COMPLETES THE TESTING OF AN RK611/RK06 SUBSYSTEM. THE DISKLESS CONTROLLER DIAGNOSTIC AND THE RK06 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 CLOCKED 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.

CAUTION

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 DRIVES
- FORMATTED RK06K ON EACH DRIVE

2.2 PRELIMINARY PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC (ALL PARTS) AND THE

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

UNIBUS RK06 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.

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

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 LOOP UNTIL A CONTROL Q (↑Q) IS TYPED.

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

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	1676
VECTOR ADDRESS	KWLVEC	1700
MM11		
UNIBUS ADDRESS		
BANK 0	MMCSR1	1702
BANK 1	MMCSR2	1704
VECTOR ADDRESS	MMVEC	1706

3.9 EXECUTION TIME

THE FIRST PASS OF THE PROGRAM FOR ONE DRIVE IS APPROXIMATELY 50 SECONDS AND EACH SUBSEQUENT PASS IS APPROXIMATELY 6 MINUTES 50 SECONDS.

THE EXECUTION TIME FOR MULTIPLE DRIVES IS THE PRODUCT OF THE NUMBER OF DRIVES TIMES 52 SECONDS FOR THE FIRST PASS. FOR SUBSEQUENT PASSES THE RUN TIME IS THE PRODUCT OF 6 MINUTES 50 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.

THE DRIVE MUST BE ON-LINE, CYCLED UP, AND WRITE ENABLED. IF ANY ONE OF THESE CONDITIONS IS NOT TRUE THAT DRIVE IS NOT TESTED AND IT IS EXPECTED TO BE OFF-LINE. ADDRESSING THAT DRIVE SHOULD CAUSE NON-EXISTANT DRIVE ERROR. 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.

TEST 4 RELEASE ALL DRIVES

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

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. 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

**WRITE HEADER TESTS

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.

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

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

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 OF 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 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

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

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.

566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620

**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

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

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

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 110
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)

677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732

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.

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

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
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

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)

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

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.

**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,

789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844

TRACK 0, SECTOR 0 WITH BUFFER BEGINNING 110 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

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:

0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
900

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

901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956

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.

TEST 75 CONTROLLER TIME OUT

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.

**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

957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012

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. 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 AVAILIABLE 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 AVAILIABLE DRIVES.

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 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 AVAILABLE DRIVES. ISSUE A SUBSYSTEM CLEAR. MAKE SURE ALL ATTENTIONS RESET.

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:

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
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
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123

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 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

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

STESTN	SERRPC	DRVNUM				
RKCS1	RKCS2	RKDS	RKER	RKASOF	RKMR1	
T.CS1	T.CS2	T.DS	T.ER	T.ASF	T.MR1	

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 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 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.

[Handwritten marks]

1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235

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

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.

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

1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291ECC 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.ASOFTHE 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 CODEDESCRIPTON OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR
ILLEGAL.

ERROR ITEM 10

(DESCRIPTION OF ERROR)
ERROR AT COMPLETION OF OPERATION
(DESCRIPITON OF OPERATION)
TST NUM ERR PC DRIVE
\$TESTN ERRPC DRVNUM
EXPT IS
\$REG10 \$REG11THE 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 RKMR1

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

T.CS1 T.CS2 T.DS T.ER T.ASOF T.MR1

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)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC 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 (TWAITN 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

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
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403

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 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.

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
1492
1493
1494
1495
1496
1497
1498
1499
1500

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

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 PHASES 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 TRANSITIONS

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
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515

.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 TOO SLOW OR
;IF HDR 0 NOT FOUND

GET DRIVE STATUS

THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN \$REG10 THROUGH \$REG17. THESE REGISTERS 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 PERIOD 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

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
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571

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

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
.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

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
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
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683

GROUP 2 ERRORS:
BIT 0 - NON-EXISTAND 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, 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.
- 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).

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
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739

- 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 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.

```

%
.NLIST CND,MD,MC,TOC
.LIST ME
.ENABL ABS,AMA
$SWR= 167400
$TN= 1
.TITLE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
;*COPYRIGHT (C) 1976
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY MARV TEGROTENHUIS
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.

```

167400
000001

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
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795


```

1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851

```

```

;*
.SBTTL OPERATIONAL SWITCH SETTINGS
;*
;* SWITCH USE
;* -----
;* 15 HALT ON ERROR
;* 14 LOOP ON TEST
;* 13 INHIBIT ERROR TIMEOUTS
;* 12 ABORT PROGRAM AFTER 20 ERRORS
;* 11 INHIBIT ITERATIONS
;* 10 BELL ON ERROR
;* 9 LOOP ON ERROR
;* 8 LOOP ON TEST IN SWR<7:0>
.SBTTL BASIC DEFINITIONS
;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS
MT= 11 ;;CODE FOR HORIZONTAL TAB
LF= 12 ;;CODE FOR LINE FEED
CR= 15 ;;CODE FOR CARRIAGE RETURN
CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776 ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774 ;;STACK LIMIT REGISTER
PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570 ;;HARDWARE SWITCH REGISTER
DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0 ;;GENERAL REGISTER
R1= %1 ;;GENERAL REGISTER
R2= %2 ;;GENERAL REGISTER
R3= %3 ;;GENERAL REGISTER
R4= %4 ;;GENERAL REGISTER
R5= %5 ;;GENERAL REGISTER
R6= %6 ;;GENERAL REGISTER
R7= %7 ;;GENERAL REGISTER
SP= %6 ;;STACK POINTER
PC= %7 ;;PROGRAM COUNTER

;*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

;* "SWITCH REGISTER" SWITCH DEFINITIONS
SW15= 100000

```


1852 040000
1853 020000
1854 010000
1855 004000
1856 002000
1857 001000
1858 000400
1859 000200
1860 000100
1861 000040
1862 000020
1863 000010
1864 000004
1865 000002
1866 000001

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
.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

1878
1879 100000
1880 040000
1881 020000
1882 010000
1883 004000
1884 002000
1885 001000
1886 000400
1887 000200
1888 000100
1889 000040
1890 000020
1891 000010
1892 000004
1893 000002
1894 000001

.*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
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907 000004

.*BASIC "CPU" TRAP VECTOR ADDRESSES
ERRVEC= 4 ;;TIME OUT AND OTHER ERRORS


```

1908      000010      RESVEC= 10      ;;RESERVED AND ILLEGAL INSTRUCTIONS
1909      000014      TBITVEC=14      ;; "T" BIT
1910      000014      TRIVEC= 14      ;; TRACE TRAP
1911      000014      BPTVEC= 14      ;; BREAKPOINT TRAP (BPT)
1912      000020      IOTVEC= 20      ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
1913      000024      PWRVEC= 24      ;; POWER FAIL
1914      000030      EMTVEC= 30      ;; EMULATOR TRAP (EMT) **ERROR**
1915      000034      TRAPVEC=34      ;; "TRAP" TRAP
1916      000060      TKVEC= 60      ;; TTY KEYBOARD VECTOR
1917      000064      TPVEC= 64      ;; TTY PRINTER VECTOR
1918      000240      PIRQVEC=240    ;; PROGRAM INTERRUPT REQUEST VECTOR
1919      .SBTTL      MEMORY MANAGEMENT DEFINITIONS
1920
1921      ;*KT11 VECTOR ADDRESS
1922
1923      000250      MMVEC= 250
1924
1925      ;*KT11 STATUS REGISTER ADDRESSES
1926
1927      177572      SR0= 177572
1928      177574      SR1= 177574
1929      177576      SR2= 177576
1930      172516      SR3= 172516
1931
1932      ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1933
1934      172300      KIPDR0= 172300
1935      172302      KIPDR1= 172302
1936      172304      KIPDR2= 172304
1937      172306      KIPDR3= 172306
1938      172310      KIPDR4= 172310
1939      172312      KIPDR5= 172312
1940      172314      KIPDR6= 172314
1941      172316      KIPDR7= 172316
1942
1943      ;*KERNEL "I" PAGE ADDRESS REGISTERS
1944
1945      172340      KIPAR0= 172340
1946      172342      KIPAR1= 172342
1947      172344      KIPAR2= 172344
1948      172346      KIPAR3= 172346
1949      172350      KIPAR4= 172350
1950      172352      KIPAR5= 172352
1951      172354      KIPAR6= 172354
1952      172356      KIPAR7= 172356
1953
1954      000210      AVECT1= 210      ;; DEFINE RK611 VECTOR INTERRUPT
1955      000240      APRIOR= PR5      ;; DEFINE RK611 PRIORITY
1956      177440      ABASE= 177440    ;; DEFINE RK611 BASE BUS ADDRESS
1957
1958      .SBTTL      RK611 CONTROLLER REGISTER DEFINITION
1959
1960      000000      RKCS1= 0      ;; CONTROL AND STATUS REGISTER 1
1961      000002      RKWC= 2      ;; WORD COUNT REGISTER
1962      000004      RKBA= 4      ;; BUS ADDRESS REGISTER
1963      000006      RKDA= 6      ;; DESIRED TRACK SECTOR REGISTER

```


1964	000010	RKCS2= 10	;CONTROL AND STATUS REGISTER 2
1965	000012	RKDS= 12	;DRIVE STATUS REGISTER
1966	000014	RKER= 14	;ERROR REGISTER
1967	000016	RKASOF= 16	;ATTENTION SUMMARY AND OFFSET REGISTER
1968	000020	RKDCYL= 20	;DESIRED CYLINDER REGISTER
1969	000024	RKDB= 24	;DATA BUFFER
1970	000026	RKMR1= 26	;MAINTENANCE REGISTER 1
1971	000034	RKMR2= 34	;MAINTENANCE REGISTER 2
1972	000036	RKMR3= 36	;MAINTENANCE REGISTER 3
1973	000030	RKECPS= 30	;ECC POSITION INFORMATION
1974	000032	RKECPT= 32	;ECC PATTERN INFORMATION
1975	000022	RKSPAR= 22	;SPARE REGISTER
1976			
1977		.SBTTL DRIVE COMMANDS	
1978			
1979	000101	SELDRV= 101	;SELECT DRIVE
1980	000103	PACK= 103	;PACK ACKNOWLEDGE
1981	000105	CLEAR= 105	;DRIVE CLEAR
1982	000107	UNLOAD= 107	;UNLOAD
1983	000111	SRTSPL= 111	;START SPINDLE
1984	000113	RECAL= 113	;RECALIBRATE
1985	000115	OFFSET= 115	;OFFSET
1986	000117	SEEK= 117	;SEEK
1987	000121	RDDATA= 121	;READ DATA
1988	000123	WRDATA= 123	;WRITE DATA
1989	000125	RDHEAD= 125	;READ HEADER
1990	000127	WRHEAD= 127	;WRITE HEADER AND DATA
1991	000131	WRTCHK= 131	;WRITE CHECK
1992	000300	INTR= 300	;GENERATE INTERRUPT TO CPU
1993			
1994		.SBTTL CONTROL AND STATUS REGISTER 1 BITS	
1995			
1996	000001	GO= BIT0	;GO BIT
1997	000100	IE= BIT6	;INTERRUPT ENABLE
1998	000200	RDY= BIT7	;CONTROLLER READY
1999	000400	BA16= BIT8	;BUS ADDRESS BIT 16
2000	001000	BA17= BIT9	;BUS ADDRESS BIT 17
2001	002000	CDT= BIT10	;CONTROLLER DRIVE TYPE (0=RK06)
2002	004000	CT0= BIT11	;CONTROLLER TIMED OUT WAITING FOR ; DRIVE RESPONSE
2003			
2004	010000	CFMT= BIT12	;CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
2005	020000	SPAR= BIT13	;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
2006	040000	DI= BIT14	;DRIVE INTERRUPT
2007	100000	CERR= BIT15	;CONTROLLER ERROR
2008	100000	CCLR= BIT15	;CONTROLLER CLEAR
2009			
2010		.SBTTL CONTROL AND STATUS REGISTER 2 BITS	
2011			
2012	000007	DRVMSK= 7	;MASK FOR DRIVE SELECTION CODE
2013	000010	RLS= BIT3	;DESELECT OR RELEASE DRIVE IN BITS 0-2
2014	000020	BAI= BIT4	;BUS ADDRESS INCREMENT INHIBIT
2015	000040	SCLR= BIT5	;CLEAR CONTROLLER AND ALL DRIVES
2016	000100	IR= BIT6	;INPUT READY
2017	000200	OR= BIT7	;OUTPUT READY
2018	000400	UFE= BIT8	;UNIT FIELD ERROR
2019	001000	MDS= BIT9	;MULTIPLE DRIVE SELECT

M03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 38
CONTROL AND STATUS REGISTER 2 BITS

SEQ 0038

2020	002000	PGE=	BIT10	; PROGRAMMING ERROR
2021	004000	NEM=	BIT11	; NON-EXISTENT MEMORY
2022	010000	NED=	BIT12	; NON-EXISTENT DRIVE
2023	020000	UPE=	BIT13	; UNIBUS PARITY ERROR
2024	040000	WCE=	BIT14	; WRITE CHECK ERROR
2025	100000	DLT=	BIT15	; DATA LATE ERROR
2026				
2027				
2028		.SBTTL	ERROR REGISTER BIT DEFINITION	
2029	000001	ILF=	BIT0	; ILLEGAL FUNCTION CODE
2030	000002	SKI=	BIT1	; SEEK INCOMPLETE
2031	000004	NXF=	BIT2	; NON-EXECUTABLE DRIVE FUNCTION
2032	000010	DRPAR=	BIT3	; DRIVE DETECTED DRIVE BUS PARITY ERROR
2033	000020	FMTE=	BIT4	; FORMAT ERROR
2034	000040	DTYPE=	BIT5	; DRIVE TYPE ERROR
2035	000100	ECH=	BIT6	; ECC HARD
2036	000200	BSE=	BIT7	; BAD SECTOR ERROR
2037	000400	HVRC=	BIT8	; HEADER VRC ERROR
2038	001000	COE=	BIT9	; CYLINDER ADDRESS OVERFLOW ERROR
2039	002000	IDAE=	BIT10	; INVALID DISK ADDRESS ERROR
2040	004000	WLE=	BIT11	; WRITE LOCK ERROR
2041	010000	DTE=	BIT12	; DRIVE TIMING ERROR
2042	020000	OPI=	BIT13	; OPERATION (SEARCH) INCOMPLETE
2043	040000	UNS=	BIT14	; DRIVE UNSAFE
2044	100000	DCK=	BIT15	; DATA CHECK
2045				
2046		.SBTTL	STATUS REGISTER BIT DEFINITION	
2047				
2048	000001	DRA=	BIT0	; DRIVE AVAILABLE (CONTROLLER IS SET IF ; THIS BIT IS RESET)
2049				
2050	000004	OFST=	BIT2	; DRIVE OFFSET
2051	000010	ACLO=	BIT3	; AC LOW
2052	000020	SPDLSS=	BIT4	; SPEED LOSS
2053	000040	DROT=	BIT5	; DRIVE OFF TRACK
2054	000100	VV=	BIT6	; VOLUME VALID
2055	000200	DRDY=	BIT7	; DRIVE READY
2056	000400	DDT=	BIT8	; DRIVE TYPE (0=RK06)
2057	004000	WRL=	BIT11	; WRITE LOCK
2058	020000	PIP=	BIT13	; POSITIONING IN PROGRESS
2059	040000	DSC=	BIT14	; DRIVE STATUS CHANGE
2060	100000	SVAL=	BIT15	; STATUS VALID
2061				
2062		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION	
2063				
2064	000017	MESMSK=	17	; MESSAGE MASK
2065				
2066	000020	PAT=	BIT4	; FORCE EVEN PARITY ON SERCON MESSAGE LINES
2067	000040	DMD=	BIT5	; DIAGNOSTIC MODE
2068	000100	MSP=	BIT6	; MAINTENANCE SECTOR PULSE
2069	000200	MIND=	BIT7	; MAINTENANCE INDEX
2070	000400	MCLK=	BIT8	; MAINTENANCE CLOCK
2071	001000	MERD=	BIT9	; MAINTENANCE ENCODED READ DATA
2072	002000	MEWD=	BIT10	; MAINTENANCE ENCODED WRITE DATA
2073	004000	PCA=	BIT11	; PRECOMPENSATION ADVANCE
2074	010000	PCD=	BIT12	; PRECOMPENSATION DELAY
2075	020000	ECCW=	BIT13	; ECC WORD IS BEING READ OR WRITTEN

2076	040000	WRTGAT= BIT14	;WRITE GATE
2077	100000	RDGATE= BIT15	;READ GATE
2078			
2079		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A
2080			
2081	000040	S.DRA= BIT5	;DRIVE AVAILIABLE
2082	000100	S.VV= BIT6	;VOLUME VALID
2083	000200	S.DRY= BIT7	;DRIVE READY
2084	000400	S.TYPE= BIT8	;DRIVE TYPE
2085	001000	S.FORM= BIT9	;DRIVE FORMAT
2086	002000	S.OFF= BIT10	;OFFSET
2087	004000	S.WRL= BIT11	;WRITE LOCK
2088	010000	S.SPIN= BIT12	;SPINDLE ON
2089	020000	S.PIP= BIT13	;POSITIONING IN PROGRESS
2090	040000	S.DSC= BIT14	;DRIVE STATUS CHANGE
2091			
2092		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B
2093			
2094	000040	S.ICYL= BIT5	;ILLEGAL CYLINDER ADDRESS
2095	000100	S.ACLO= BIT6	;AC LOW
2096	000200	S.FLT= BIT7	;DRIVE FAULT
2097	000400	S.ILF= BIT8	;ILLEGAL FUNCTION
2098	001000	S.PAR= BIT9	;DRIVE DETECTED SERCON PARITY ERROR
2099	002000	S.SKI= BIT10	;SEEK INCOMPLETE
2100	004000	S.WLE= BIT11	;WRITE LOCK ERROR
2101	010000	S.SPLS= BIT12	;SPEED LOSS
2102	020000	S.DROT= BIT13	;DRIVE OFF TRACK
2103	040000	S.UNS= BIT14	;DRIVE UNSAFE
2104			
2105		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
2106			
2107	000020	S.XDOK= BIT4	;TRANSDUCER OK
2108	000040	S.HDHM= BIT5	;HEADS HOME
2109	000100	S.BRHM= BIT6	;BRUSHES HOME
2110	000200	S.DOOR= BIT7	;DOOR INTERLOCKED
2111	000400	S.CART= BIT8	;CARTRAGE INTERLOCK
2112	001000	S.SPOK= BIT9	;SPEED OK
2113	002000	S.FWD= BIT10	;FORWARD
2114	004000	S.REV= BIT11	;REVERSE
2115	010000	S.LOAD= BIT12	;HEADS LOADING
2116	020000	S.RTZ= BIT13	;RETURN TO ZERO
2117	040000	S.UNLD= BIT14	;HEADS UNLOADING
2118			
2119		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
2120			
2121	000020	S.SECT= BIT4	;SECTOR ERROR
2122	000040	S.WCLK= BIT5	;WRITE CLOCK AND NO WRITE GATE
2123	000100	S.WGAT= BIT6	;WRITE GATE AND NO TRANSISTIONS
2124	000200	S.HDFL= BIT7	;HEAD FAULT
2125	000400	S.MHD= BIT8	;MULTIPLE HEAD SELECT
2126	001000	S.XERR= BIT9	;INDEX ERROR
2127	002000	S.DIB= BIT10	;DIBIT ERROR
2128	004000	S.PLO= BIT11	;PLO ERROR
2129	010000	S.NMOV= BIT12	;SEEK AND NO MOTION
2130	020000	S.LIMD= BIT13	;LIMIT DETECT ON SEEK
2131	040000	S.BRAKE= BIT14	;SERVO-BRAKE

2188		
2189		
2190	000224	
2191	000224	000000
2192	000226	001276
2193	000230	000024
2194	000232	000074
2195	000234	000740
2196	000236	000031

;INTERFACE SPEC.

\$APTHD:			
\$HIPTS:	.WORD	0	:: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
\$MBADR:	.WORD	\$MAIL	:: ADDRESS OF APT MAILBOX (BITS 0-15)
\$TSTM:	.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)

.SBTTL COMMON TAGS

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

2197
2198
2199
2200
2201
2202
2203 001100
2204 001100
2205 001100 000000
2206 001102 000
2207 001103 000
2208 001104 000000
2209 001106 000000
2210 001110 000000
2211 001112 000000
2212 001114 000
2213 001115 001
2214 001116 000000
2215 001120 000000
2216 001122 000000
2217 001124 000000
2218 001126 000000
2219 001130 000000
2220 001132 000000
2221 001134 000
2222 001135 000
2223 001136 000000
2224 001140 177570
2225 001142 177570
2226 001144 177560
2227 001146 177562
2228 001150 177564
2229 001152 177566
2230 001154 000
2231 001155 002
2232 001156 012
2233 001157 000
2234 001160 000000
2235
2236 001162 000000
2237 001164 000000
2238 001166 000000
2239 001170 000000
2240 001172 000000
2241 001174 000000
2242 001176 000000
2243 001200 000000
2244 001202 000000
2245 001204 000000
2246 001206 000000
2247 001210 000000
2248 001212 000000
2249 001214 000000
2250 001216 000000
2251 001220 000000
2252 001222 000000

SCMTAG: . =1100
\$STNM: .WORD 0
\$ERFLG: .BYTE 00
\$ICNT: .WORD 00
\$LPADR: .WORD 00
\$LPERR: .WORD 00
\$ERTTL: .WORD 00
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 00
\$BDADR: .WORD 00
\$GDDAT: .WORD 00
\$BDDAT: .WORD 00
\$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 00
\$REG2: .WORD 00
\$REG3: .WORD 00
\$REG4: .WORD 00
\$REG5: .WORD 00
\$REG6: .WORD 00
\$REG7: .WORD 00
\$REG10: .WORD 00
\$REG11: .WORD 00
\$REG12: .WORD 00
\$REG13: .WORD 00
\$REG14: .WORD 00
\$REG15: .WORD 00
\$REG16: .WORD 00
\$REG17: .WORD 00
\$TMPD: .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

2253	001224	000000	\$TMP1:	.WORD	0	::USER DEFINED	
2254	001226	000000	\$TMP2:	.WORD	00	::USER DEFINED	
2255	001230	000000	\$TMP3:	.WORD	00	::USER DEFINED	
2256	001232	000000	\$TMP4:	.WORD	00	::USER DEFINED	
2257	001234	000000	\$TMP5:	.WORD	00	::USER DEFINED	
2258	001236	000000	\$TMP6:	.WORD	00	::USER DEFINED	
2259	001240	000000	\$TMP7:	.WORD	00	::USER DEFINED	
2260	001242	000000	\$TMP10:	.WORD	00	::USER DEFINED	
2261	001244	000000	\$TMP11:	.WORD	00	::USER DEFINED	
2262	001246	000000	\$TMP12:	.WORD	00	::USER DEFINED	
2263	001250	000000	\$TMP13:	.WORD	00	::USER DEFINED	
2264	001252	000000	\$TMP14:	.WORD	00	::USER DEFINED	
2265	001254	000000	\$TMP15:	.WORD	00	::USER DEFINED	
2266	001256	000000	\$TMP16:	.WORD	00	::USER DEFINED	
2267	001260	000000	\$TMP17:	.WORD	0	::USER DEFINED	
2268	001262	000000	\$TIMES:	0		::MAX. NUMBER OF ITERATIONS	
2269	001264	000000	\$ESCAPE:	0		::ESCAPE ON ERROR ADDRESS	
2270	001266	177607	\$BELL:	.ASCIZ	<207><377><377>	::CODE FOR BELL	
2271	001272	077	\$QUES:	.ASCII	??	::QUESTION MARK	
2272	001273	015	\$CRLF:	.ASCII	<15>	::CARRIAGE RETURN	
2273	001274	000012	\$LF:	.ASCIZ	<12>	::LINE FEED	
2274			:*****				
2275			.SBTTL APT MAILBOX-ETABLE				
2276			:*****				
2277			.EVEN				
2278			\$MAIL:			::APT MAILBOX	
2279	001276		\$MSGTY:	.WORD	AMSGTY	::MESSAGE TYPE CODE	
2280	001276	000000	\$FATAL:	.WORD	AFATAL	::FATAL ERROR NUMBER	
2281	001300	000000	\$TESTN:	.WORD	ATESTN	::TEST NUMBER	
2282	001302	000000	\$PASS:	.WORD	APASS	::PASS COUNT	
2283	001304	000000	\$DEVCT:	.WORD	ADEVCT	::DEVICE COUNT	
2284	001306	000000	\$UNIT:	.WORD	AUNIT	::I/O UNIT NUMBER	
2285	001310	000000	\$MSGAD:	.WORD	AMSGAD	::MESSAGE ADDRESS	
2286	001312	000000	\$MSGLG:	.WORD	AMSGLG	::MESSAGE LENGTH	
2287	001314	000000	\$ETABLE:			::APT ENVIRONMENT TABLE	
2288	001316		\$ENV:	.BYTE	AENV	::ENVIRONMENT BYTE	
2289	001316	000	\$ENVM:	.BYTE	AENVM	::ENVIRONMENT MODE BITS	
2290	001317	000	\$SWREG:	.WORD	ASWREG	::APT SWITCH REGISTER	
2291	001320	000000	\$USWR:	.WORD	AUSWR	::USER SWITCHES	
2292	001322	000000	\$CPUOP:	.WORD	ACPUOP	::CPU TYPE, OPTIONS	
2293	001324	000000	:*				BITS 15-11=CPU TYPE
2294			:*				11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
2295			:*				11/70=06, PDQ=07, Q=10
2296			:*				BIT 10=REAL TIME CLOCK
2297			:*				BIT 9=FLOATING POINT PROCESSOR
2298			:*				BIT 8=MEMORY MANAGEMENT
2299			:*				::HIGH ADDRESS, M.S. BYTE
2300	001326	000	\$MAMS1:	.BYTE	AMAMS1	::MEM. TYPE, BLK#1	
2301	001327	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE BYTE -- (HIGH BYTE)	
2302			:*				900 NSEC CORE=001
2303			:*				300 NSEC BIPOLAR=002
2304			:*				500 NSEC MOS=003
2305			:*				::HIGH ADDRESS, BLK#1
2306	001330	000000	\$MADR1:	.WORD	AMADR1	MEM.LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE	
2307			:*				::HIGH ADDRESS, M.S. BYTE
2308	001332	000	\$MAMS2:	.BYTE	AMAMS2		

2309	001333	000	\$MTYP2: .BYTE	AMTYP2	:: MEM. TYPE, BLK#2
2310	001334	000000	\$MADR2: .WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
2311	001336	000	\$MAMS3: .BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
2312	001337	000	\$MTYP3: .BYTE	AMTYP3	:: MEM. TYPE, BLK#3
2313	001340	000000	\$MADR3: .WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
2314	001342	000	\$MAMS4: .BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
2315	001343	000	\$MTYP4: .BYTE	AMTYP4	:: MEM. TYPE, BLK#4
2316	001344	000000	\$MADR4: .WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
2317	001346	000210	\$VECT1: .WORD	AVECT1	:: INTERRUPT VECTOR#1, BUS PRIORITY#1
2318	001350	000000	\$VECT2: .WORD	AVECT2	:: INTERRUPT VECTOR#2, BUS PRIORITY#2
2319	001352	177440	\$BASE: .WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
2320	001354	000000	\$DEVM: .WORD	ADEVM	:: DEVICE MAP
2321	001356	000000	\$CDW1: .WORD	ACDW1	:: CONTROLLER DESCRIPTION WORD#1
2322	001360		\$ETEND:		
2323			.MEXIT		

2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
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

001360

001360 000000
001362 056027
001364 057400
001366 057472

.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

\$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 WORK 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
:* SIGHTLY 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

EM1N: .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

2380
2381
2382
2383
2384 001370 000000
2385 001372 000000
2386 001374 057406
2387 001376 057476
2388
2389
2390
2391
2392
2393
2394
2395 001400 000000
2396 001402 056055
2397 001404 057356
2398 001406 057516
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
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

```

:*      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  RKMR1
:*      T.CS1  T.CS2  T.DS   T.ER   T.AST  T.MR1
EM3N:   .WORD   0
DH002A  DT003
        DF003

```

```

:* ERROR ITEMS 4,5,6,8,7 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.
:*

```


2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
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

:* 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 057406
001416 057526

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

2492 001424 057406
2493 001426 057556
2494
2495
2496
2497
2498
2499

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

2515 001430 000000
2516 001432 000000
2517 001434 057406
2518 001436 057612
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540

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

2541 001440 000000
2542 001442 000000
2543 001444 057406
2544 001446 057636
2545
2546
2547

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

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


```

2548      :*      ERROR AT COMPLETION OF OPERATION
2549      :*      (DESCRIPITON OF OPERATION)
2550      :*      TST NUM ERR PC DRIVE
2551      :*      $TESTN $ERRPC DRVNUM
2552      :*      EXPT 15
2553      :*      $REG10 $REG11
2554
2555      :*      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
2556      :*      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS.  THE SPECIFIC
2557      :*      ERRORS ARE:
2558      :*      WORD COUNT INCORRECT
2559      :*      BUS ADDRESS INCORRECT
2560      :*      CYLINDER ADDRESS INCORRECT
2561      :*      TRACK ADDRESS INCORRECT
2562      :*      SECTOR ADDRESS INCORRECT
2563
2564      001450 000000      EM10N: .WORD 0
2565      001452 056670      DHD10
2566      001454 057456      DTD15
2567      001456 057656      DFD10
2568
2569      :*      ERROR ITEM 11
2570      :*      (ERROR INDICATOR OR STATUS BIT)
2571      :*      NOT SET AS A RESULT OF
2572      :*      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
2573      :*      TST NUM ERR PC DRIVE
2574      :*      $TESTN $ERRPC DRVNUM
2575      :*      RKCS1  RKCS2  RKDS  RKER  RKASOF  RKMR1
2576      :*      T.CS1  T.CS2  T.DS  T.ER  T.ASOF  T.MR1
2577
2578      001460 000000      EM11N: .WORD 0
2579      001462 057014      DHD11
2580      001464 057406      DTD10
2581      001466 057676      DFD11
2582
2583      :*      ERROR ITEM 12
2584      :*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2585      :*      "NOT RESET AS A RESULT OF"
2586
2587      001470 000000      EM12N: .WORD 0
2588      001472 057043      DHD12
2589      001474 057406      DTD10
2590      001476 057676      DFD11
2591
2592      :*      ERROR ITEM 13
2593      :*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2594      :*      "SET AS A RESULT OF"
2595
2596      001500 000000      EM13N: .WORD 0
2597      001502 057074      DHD13
2598      001504 057406      DTD10
2599      001506 057676      DFD11
2600
2601      :*      ERROR ITEM 14
2602      :*      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2603      :*      "RESET AS A RESULT OF"

```


2604	001510	000000
2605	001512	057117
2606	001514	057406
2607	001516	057676
2608		
2609		
2610		
2611		
2612		
2613		
2614		
2615		
2616	001520	000000
2617	001522	056055
2618	001524	057456
2619	001526	057720
2620		
2621		
2622		
2623		
2624	001530	000000
2625	001532	000000
2626	001534	057464
2627	001536	057730
2628		

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


```

2629          .SBTTL REGISTER STORAGE FOR TEST
2630
2631 001540 000000 T.CS1: .WORD 0
2632 001542 000000 T.WC: .WORD 0
2633 001544 000000 T.BA: .WORD 0
2634 001546 000000 T.DA: .WORD C
2635 001550 000000 T.CS2: .WORD 0
2636 001552 000000 T.DS: .WORD 0
2637 001554 000000 T.ER: .WORD 0
2638 001556 000000 T.ASOF: .WORD 0
2639 001560 000000 T.DCYL: .WORD 0
2640 001562 000000 T.SPAR: .WORD 0
2641 001564 000000 T.DB: .WORD 0
2642 001566 000000 T.MR1: .WORD 0
2643 001570 000000 T.ECPS: .WORD 0
2644 001572 000000 T.ECPT: .WORD 0
2645 001574 000000 T.MR2: .WORD 0
2646 001576 000000 T.MR3: .WORD 0
2647
2648          .SBTTL REGISTER SETUP STORAGE
2649 001600 000100 L.CS1: .WORD 100 ;PRESET WITH INTERRUPT ENABLE
2650 001602 000000 L.WC: .WORD 0
2651 001604 000000 L.BA: .WORD 0
2652 001606          L.DA:
2653 001606          000 L.DS: .BYTE 0
2654 001607          000 L.DT: .BYTE 0
2655 001610 000000 L.CS2: .WORD 0
2656 001612 000000 L.ASOF: .WORD 0
2657 001614 000000 L.DCYL: .WORD 0
2658 001616 000000 L.MR1: .WORD 0
2659          .SBTTL PROGRAM DEFINED VARIABLES
2660
2661 001620 000000 RKVEC: .WORD 0 ;RK VECTOR
2662 001622 000000 RKPRI: .WORD 0 ;RK PRIORITY
2663 001624 000000 SRTFLG: .WORD 0 ;START FLAG
2664          ; 0 = 200
2665          ; 1 = 214
2666          ; -1 = 204
2667 001626 000000 DRVNUM: .WORD 0 ;DRIVE UNDER TEST
2668 001630 000000 DRVBIT: .WORD 0 ;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
2669 001632 000024 ERRCNT: .WORD ↑D20 ;ERROR COUNTER TO LIMIT ERROR
2670          ; ERRORS REPORTED IN PROGRAM
2671 001634 000024 ERRLMT: .WORD ↑D20 ;DATA COMPARE ERROR LIMIT
2672 001636 060060 BSF24P: .WORD BS24 ;POINTER TO BAD SECTORS 24 SECTOR MODE
2673          ; (FACTORY)
2674 001640 057734 BSF26P: .WORD BS26 ;POINTER TO BAD SECTORS 26 SECTOR MODE
2675          ; (FACTORY)
2676 001642 000000 BSS24P: .WORD 0 ;POINTER TO BAD SECTORS 24 SECT MODE
2677          ; (SOFTWARE)
2678 001644 000000 BSS26P: .WORD 0 ;POINTER TO BAD SECTORS 26 SECTOR MODE
2679          ; (SOFTWARE)
2680 001646 000000 BS26CT: .WORD 0 ;COUNT OF BAD SECTORS 26 SECTOR MODE
2681 001650 000000 BS24CT: .WORD 0 ;COUNT OF BAD SECTORS 24 SECTOR MODE
2682 001652 000764 MILCNT: .WORD ↑D500 ;COUNT TO APPROXIMATE 1 MILL SEC
2683 001654 000017 TIMCNT: .WORD ↑D15 ;COUNTER FOR MYTIME ROUTINE
2684 001656 000000 OPTFLG: .WORD 0 ;OPTION FLAGS

```


2685					
2686		000001		DOTST=	BIT0
2687		000002		MEMSZB=	BIT1
2688		000004		MEMPYB=	BIT2
2689		000010		SRTINS=	BIT3
2690		000200		PARBK0=	BIT7
2691		000100		PARBK1=	BIT6
2692		000200		BSEPT=	BIT7
2693		000400		FPFMT=	BIT8
2694		100000		LCLKPR=	BIT15
2695					
2696	001660	000000	LCLKTK:	.WORD	0
2697	001662	000000	INTSET:	.WORD	0
2698					
2699	001664	000000	REFMT:	.WORD	0
2700					
2701			:	THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR	
2702			:	HEADER OR DATA COMPARE CONTINUATION PROCESS.	
2703	001666	000000	DESOLD:	.WORD	0
2704	001670	000000	SACHLD:	.WORD	0
2705	001672	000000	WRDNUM:	.WORD	0
2706	001674	000000	WRDCNT:	.WORD	0
2707	001676	177546	KWLADD:	.WORD	177546
2708	001700	000100	KWLVEC:	.WORD	100
2709	001702	172100	MMCSR1:	.WORD	172100
2710	001704	172102	MMCSR2:	.WORD	172102
2711	001706	000114	MMVECA:	.WORD	114

;DRIVE 0 TO BE TESTED FLAG
;MEMORY SIZE REPORT FLAG
;MEMORY PARITY REPORT FLAG
;START UP INSTRUCTIONS REPORTED FLAG
;PARITY OPTION BANK 0
;PARITY OPTION BANK 0 INTERLEAVED MEM
;BSE HAS BEEN REPORTED
;FIRST PASS FORMAT SWITCH
;LINE CLOCK PRESENT
;LINE CLOCK TICK COUNTER
;NON-ZERO IF RK06 INTERRUPT SINCE LAST
;CLEAR
;REFORMAT SWITCH FOR HALT
;DESTINATION HOLD
;SOURCE HOLD
;WORD NUMBER IN ERROR HOLD
;WORDS LEFT IN COMPARE HOLD
;KW11-L ADDRESS
;KW11-L VECTOR
;MM11 ADDRESS
;MM11 ADDRESS
;MM11 VECTOR


```

2712
2713
2714
2715
2716
2717
2718 001710 012737 000002 001624 SETCLK: MOV #2,SRTFLG ;SET START FLAG FOR CLOCK ADJUST
2719 001716 000412 BR START1
2720
2721 001720 012737 000001 001624 PARM: MOV #1,SRTFLG ;SET START FLAG FOR PARMETER START
2722 001726 000406 BR START1
2723
2724 001730 012737 177777 001624 RESTRT: MOV #-1,SRTFLG ;LOAD START FLAG FOR PARMETER START
2725 001736 000402 BR START1
2726
2727 001740 005037 001624 START: CLR SRTFLG ;CLEAR START FLAG
2728 001744 000005 START1: RESET ;RESET THE WHOLE SYSTEM
2729 001746 012706 001100 MOV #STACK,SP ;INITIALIZE STACK POINTER
2730 001752 012746 000340 MOV #PR7,-(SP) ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
2731 001756 012746 001764 MOV #1$,-(SP) ;LOAD START OF PROGRAM
2732 001762 000002 RTI ;LOAD PSW
2733
2734 001764 004737 043664 1$: JSR PC,STKINT ;INITIALIZE KEYBOARD
2735 001770 005037 001664 CLR REFORMAT ;CLEAR REFORMAT SWITCH
2736
2737 .SBTTL INITIALIZE THE COMMON TAGS
2738 001774 012706 001100 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2739 002000 005026 MOV #SCMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
2740 002002 022706 001140 CLR (R6)+ ;;CLEAR MEMORY LOCATION
2741 002006 001374 CMP #SWR,R6 ;;DONE?
2742 002010 012706 001100 BNE -6 ;;LOOP BACK IF NO
2743 MOV #STACK,SP ;;SETUP THE STACK POINTER
2744 ;;INITIALIZE A FEW VECTORS
2745 002014 012737 031760 000020 MOV #$$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2746 002022 012737 000340 000022 MOV #340,@#IOTVEC+2 ;;LEVEL 7
2747 002030 012737 033000 000030 MOV #SERAR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2748 002036 012737 000340 000032 MOV #340,@#EMTVEC+2 ;;LEVEL 7
2749 002044 012737 045524 000034 MOV #STRAP,@#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
2750 002052 012737 000340 000036 MOV #340,@#TRAPVEC+2;LEVEL 7
2751 002060 012737 045346 000024 MOV #SPWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2752 002066 012737 000340 000026 MOV #340,@#PWRVEC+2 ;;LEVEL 7
2753 002074 013737 031276 031270 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
2754 002102 005037 001262 CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
2755 002106 005037 001264 CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2756 002112 112737 000001 001115 MOVB #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
2757 002120 012737 002120 001106 MOV #.,$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2758 002126 012737 002126 001110 MOV #.,$LPERR ;;SETUP THE ERROR LOOP ADDRESS
2759 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2760 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
2761 002134 013746 000004 MOV @#ERRVEC,-(SP) ;;SAVE ERROR VECTOR
2762 002140 012737 002174 000004 MOV #64$,@#ERRVEC ;;SET UP ERROR VECTOR
2763 002146 012737 177570 001140 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
2764 002154 012737 177570 001142 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
2765 002162 022777 177777 176750 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
2766 002170 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2767 002172 000403 BR 65$ ;;AND THE HARDWARE SWR IS NOT = -1
;;BRANCH IF NO TIMEOUT

```



```

2768 002174 012716 002202      64$:  MOV    #65$, (SP)      ;;SET UP FOR TRAP RETURN
2769 002200 000002                    RTI
2770 002202 012737 000176 001140 65$:  MOV    #SWREG, SWR      ;;POINT TO SOFTWARE SWR
2771 002210 012737 000174 001142  MOV    #DISPREG, DISPLAY
2772 002216 012637 000004      66$:  MOV    (SP)+, @#ERRVEC  ;;RESTORE ERROR VECTOR
2773
2774 002222 005037 001304                    CLR    $PASS             ;;CLEAR PASS COUNT
2775 002226 132737 000200 001317  BITB   #APTSIZE, $ENVM   ;;TEST USER SIZE UNDER APT
2776 002234 001403                    BEQ    67$              ;;YES, USE NON-APT SWITCH
2777 002236 012737 001320 001140  MOV    #SSWREG, SWR     ;;NO, USE APT SWITCH REGISTER
2778 002244
2779
2780                                .SBTTL  TYPE PROGRAM NAME
2781 002244 005227 177777      ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2782 002250 001066                    INC    #-1              ;;FIRST TIME?
2783 002252 022737 031432 000042  BNE    68$              ;;BRANCH IF NO
2784 002260 001462                    CMP    #SENDAD, @#42   ;;ACT-11?
2785 002262 104401 002330  BEQ    68$              ;;BRANCH IF YES
2786                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2787 002266 005737 000042  TST    @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2788 002272 001012                    BNE    70$              ;;BRANCH IF YES
2789 002274 123727 001316 000001  CMPB   $ENV, #1        ;;ARE WE RUNNING UNDER APT?
2790 002302 001406                    BEQ    70$              ;;BRANCH IF YES
2791 002304 023727 001140 000176  CMP    SWR, #SWREG     ;;SOFTWARE SWITCH REG SELECTED?
2792 002312 001005                    BNE    71$              ;;BRANCH IF NO
2793 002314 104406                    GTSWR                    ;;GET SOFT-SWR SETTINGS
2794 002316 000403                    BR     71$
2795 002320 112737 000001 001134 70$:  MOVB   #1, $AUTOB     ;;SET AUTO-MODE INDICATOR
2796 002326
2797 002326 000437      71$:
2798                                ;;69$:  .ASCIZ  <CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC  MAINDEC-11-DZR6K-C*<CRLF>
2799 002426
2800 002426 132737 000200 001317 68$:  BITB   #BIT7, $ENVM   ;;TEST IF DO NOT SIZE
2801 002434 001043                    BNE    3$              ;;YES - SKIP
2802 002436 004737 031452                    JSR    PC, $SIZE
2803 002442 023727 031756 000740  CMP    $LSTBK, #740    ;;MAKE SURE MEMORY IS SUFFICIENT
2804 002450 103007                    BHIS   2$              ;;YES - SKIP
2805 002452 104401 047613                    TYPE   ,OPROOS        ;;MESSAGE (NOT ENOUGH MEMORY)
2806 002456 012737 000001 031270  MOV    #1, $EOPCT     ;;FORCE END OF PROGRAM
2807 002464 000137 031242                    JMP    $EOP
2808 002470 013700 031756      2$:  MOV    $LSTBK, R0     ;;GET LAST BANK
2809 002474 012701 000006                    MOV    #6, R1         ;;SET SHIFT COUNT
2810 002500 013703 031754                    MOV    $LSTAD, R3    ;;GET LAST ADDRESS
2811 002504 005004                    CLR    R4             ;;CLEAR R4 FOR OVERFLOW
2812 002506 005737 031510                    TST    $KT11         ;;MEM MANAGE PRESENT?
2813 002512 100005                    BPL    23$           ;;NO - SKIP
2814 002514 006100      22$:  ROL    R0            ;;SHIFT BANK LEFT
2815 002516 006104                    ROL    R4            ;;ADD IN CARRY
2816 002520 005301                    DEC    R1            ;;DECREMENT COUNT
2817 002522 001374                    BNE    22$          ;;LOOP IF NOT ZERO
2818 002524 050003                    BIS    R0, R3        ;;SET BANK BITS IN LAST ADDRESS
2819 002526 112737 000001 001327 23$:  MOVB   #1, $MTYP1    ;;FORCE MEMORY TYPE TO 1
2820 002534 110437 001326                    MOVB   R4, $MAMS1    ;;STORE UPPER MEMORY ADDRESS
2821 002540 010337 001330                    MOV    R3, $MADR1    ;;STORE LOWER ADDRESS
2822 002544 032737 000010 001656 3$:  BIT    #$RTINS, $OPTFLG ;;TEST IF ALREDY REPORTED
2823 002552 001005                    BNE    24$          ;;YES - SKIP

```


2824	002554	104401	050604			TYPE	OPR016	;TYPE STARTUP INSTRUCTIONS
2825	002560	052737	000010	001656		BIS	#SRTINS,OPTFLG	;SET REPORTED FLAG
2826	002566				24\$:			
2827	002566	022737	000001	001624		CMP	#1 SRTFLG	;CHECK IF PARAMETER START
2828	002574	001122				BNE	15\$;NO. START TESTING
2829	002576	104401	047504		5\$:	TYPE	OPR001	;TYPE "RK611 BUS ADDRESS () ="
2830	002602	013746	001352			MOV	\$BASE,-(SP)	::SAVE \$BASE FOR TYPEOUT
2831	002606	104402				TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
2832	002610	104401	047533			TYPE	,OPR002	
2833	002614	104412				RDOCT		;GET VALUE
2834	002616	012637	001222			MOV	(SP)+,\$TMPD	
2835	002622	001407				BEQ	7\$;CHECK IF <CR>
2836	002624	022737	160000	001222		CMP	#160000,\$TMPD	;CHECK IF IN I/O PAGE
2837	002632	101361				BHI	5\$	
2838	002634	013737	001222	001352		MOV	\$TMPD,\$BASE	;LOAD NEW BUS ADDRESS
2839	002642	104401	047541		7\$:	TYPE	OPR003	;TYPE "RK611 VECTOR ADDRESS () ="
2840	002646	013746	001346			MOV	\$VECT1,-(SP)	;GET \$VECT1 FOR TYPEOUT
2841	002652	042716	160000			BIC	#160000,(SP)	;CLEAR PRIORITY BITS
2842	002656	104402				TYPOC		
2843	002660	104401	047533			TYPE	,OPR002	
2844	002664	104412				RDOCT		;GET VALUE
2845	002666	012637	001222			MOV	(SP)+,\$TMPD	
2846	002672	001412				BEQ	10\$;CHECK IF <CR>
2847	002674	022737	001000	001222		CMP	#1000,\$TMPD	
2848	002702	101757				BLOS	7\$;CHECK IF LEGAL
2849	002704	042737	017777	001346		BIC	#17777,\$VECT1	;CLEAR OLD VECTOR
2850	002712	053737	001222	001346		BIS	\$TMPD,\$VECT1	;LOAD NEW VECTOR ADDRESS
2851	002720	104401	047571		10\$:	TYPE	,OPR004	;TYPE "RK611 PRIORITY () ="
2852	002724	005046				CLR	-(SP)	
2853	002726	113716	001347			MOVB	\$VECT1+1,(SP)	
2854	002732	006216				ASR	(SP)	;SHIFT 5 BITS RIGHT
2855	002734	006216				ASR	(SP)	
2856	002736	006216				ASR	(SP)	
2857	002740	006216				ASR	(SP)	
2858	002742	006216				ASR	(SP)	
2859	002744	104402				TYPOC		
2860	002746	104401	047533			TYPE	,OPR002	
2861	002752	104412				RDOCT		;GET VALUE
2862	002754	012637	001222			MOV	(SP)+,\$TMPD	
2863	002760	001430				BEQ	15\$;CHECK IF <CR>
2864	002762	022737	000007	001222		CMP	#7,\$TMPD	;CHECK IF LEGAL
2865	002770	103753				BLO	10\$	
2866	002772	022737	000004	001222		CMP	#4,\$TMPD	
2867	003000	101347				BHI	10\$	
2868	003002	006337	001222			ASL	\$TMPD	;SHIFT 5 BITS LEFT
2869	003006	006337	001222			ASL	\$TMPD	
2870	003012	006337	001222			ASL	\$TMPD	
2871	003016	006337	001222			ASL	\$TMPD	
2872	003022	006337	001222			ASL	\$TMPD	
2873	003026	042737	160000	001347		BIC	#160000,\$VECT1+1	;CLEAR OLD PRIORITY
2874	003034	053737	001222	001347		BIS	\$TMPD,\$VECT1+1	;LOAD RK611 PRIORITY
2875	003042	004737	033704		15\$:	JSR	PC,OPTTST	;SETUP PARITY CHECK & CLOCK
2876	003046	013700	001346			MOV	\$VECT1,RO	;STORE VECTOR FOR USE
2877	003052	042700	160000			BIC	#160000,RO	;CLEAR PRIORITY BITS
2878	003056	010037	001620			MOV	RO,RKVEC	
2879	003062	012710	033626			MOV	#INTHLR,(RO)	;SETUP INTERRUPT ADDRESS

E05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 56
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0056

2880	003066	113737	001347	001622	MOVB	\$VECT1+1,RKPRI	:STORE PRIORITY FOR USE
2881	003074	013702	001352		MOV	\$BASE,R2	:SET BASE ADDRESS
2882	003100	005037	00:264		CLR	\$ESCAPE	:CLEAR ESCAPE
2883	003104	012746	000000		MOV	#PRD,-(SP)	:SET PRIORITY
2884	003110	012746	003116		MOV	#16\$,-(SP)	
2885	003114	000002			RTI		
2886	003116			16\$:			
2887							


```

2888
2889
2890
2891
2892
2893
2894
2895
2896 003116 000004
2897 003120 012737 000100 001262
2898 003126 012706 001100
2899 003132 012701 000004
2900 003136 012146
2901 003140 011146
2902 003142 012701 000004
2903 003146 012721 033620
2904 003152 012711 000340
2905 003156 013702 001352
2906 003162 005037 001662
2907 003166 012762 000000 000000
2908 003174 000240
2909 003176 000240
2910 003200 000240
2911 003202 005737 001662
2912 003206 001406
2913 003210 012737 052216 001360
2914 003216 104001
2915 003220 000137 042442
2916 003224 012701 000006
2917 003230 012611
2918 003232 012641
2919
2920
2921
2922
2923
2924
2925
2926
2927 003234 000004
2928 003236 012737 000100 001262
2929 003244 012762 005000 000010
2930
2931 003252 005037 001662
2932 003256 012762 000300 000000
2933 003264 000240
2934 003266 000240
2935 003270 000240
2936 003272 005737 001662
2937 003276 001011
2938 003300 105737 001103
2939
2940
2941 003304 001004
2942 003306 012737 052216 001360
2943 003314 104001

```

```

.SBTTL **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. IF A TRAP OCCURS
:* THE PROGRAM IS HALTED.
:*****
TST1: SCOPE
MOV #100,$TIMES ;:DO 100 ITERATIONS
MOV #STACK,SP ;:CLEAN OFF STACK
MOV #4,R1 ;:SET POINTER TO VECTOR
MOV (R1)+,-(SP) ;:STORE OLD VECTOR CONTENTS
MOV (R1)-,(SP)
MOV #4,R1 ;:RESET POINTER
MOV #NEXINT,(R1)+ ;:SET VECTOR TO NEM TEST HANDLER
MOV #PR7,(R1) ;:SET PRIORITY
MOV $BASE,R2 ;:SET POINTER TO RK611 BASE ADDRESS
CLR INTSET ;:CLEAR INTERRUPT COUNTER
MOV #0,RKCS1(R2) ;:WRITE CS1 TO SEE IN NEM WILL SET
NOP
NOP
NOP
TST INTSET ;:TEST IF COUNTER IS 0
BEQ 1$ ;:YES - SKIP ERROR REPORT
MOV #EM1,EMIN ;:MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
ERROR 1
JMP CTRHLT ;:GO TO CONTROLLED HALT
1$: MOV #6,R1 ;:RESTORE VECTOR
MOV (SP)+,(R1)
MOV (SP)+,-(R1)
:*****
:*TEST 2 INTERRUPT VECTOR ADDRESS TEST
:* CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
:* EXPECTED ADDRESS. IF INTERRUPT VECTOR IS IN ERROR,
:* THE PROGRAM IS HALTED.
:*****
TST2: SCOPE
MOV #100,$TIMES ;:DO 100 ITERATIONS
MOV #CLR,RKCS2(R2) ;:CLEAR SUBSYSTEM, SPECIFICALLY TO
;:CLEAR ANY OLD INTERRUPTS
CLR INTSET ;:CLEAR INTERRUPT COUNTER
MOV #RDY!IE,RKCS1(R2) ;:WRITE CS1 TO FORCE INTERRUPT
NOP
NOP
NOP
TST INTSET ;:TEST IF INTERRUPT OCCURRED
BNE 3$ ;:YES - SKIP ERROR REPORT
TSTB $ERFLG ;:TEST IF ERFLG ALREADY SET. IF SO THE
;:INTERRUPT WENT TO THE WRONG VECTOR
;:AND MESSAGE HAS BEEN REPORTED.
BNE 2$ ;:THEREFORE - EXIT
MOV #EM1,EMIN ;:MESSAGE (NO INTERRUPT)
ERROR 1

```


2944 003316 000137 042442
2945 003322

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.

* THE DRIVE MUST BE ON-LINE, CYCLED UP, AND WRITE ENABLED.
* IF ANY ONE OF THESE CONDITIONS IS NOT TRUE THAT DRIVE
* IS NOT TESTED AND IT IS EXPECTED TO BE OFF-LINE. ADDRESSING
* THAT DRIVE SHOULD CAUSE NON-EXISTANT DRIVE ERROR.
* 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.

2987 003322 000004
2988 003324 012737 000062 001262
2989 003332 104416
2990 003334 104003
2991
2992 003336 012746 000000
2993 003342 012746 003350
2994 003346 000002
2995
2996 003350 013701 001620
2997 003354 012721 033626
2998 003360 012711 000340
2999 003364 012703 001354

TST3: SCOPE
MOV #50, \$TIMES ;:DO 50. ITERATIONS
TSSINIT ;CALL SUBSYSTEM CLEAR AND TEST
ERROR 3
MOV #PRO, -(SP) ;SET PROCESSOR PRIORITY TO ALLOW
MOV #1\$, -(SP) ;RK611 INTERRUPTS
RTI
1\$: MOV RKVEC, R1 ;GET VECTOR
MOV #INTHLR, (R1)+ ;LOAD INTERRUPT VECTOR
MOV #PR7, (R1)
MOV #SDEV, R3 ;GET ADDRESS OF DEVICE MAP

H05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 59
T3 SELECT ALL DRIVES

SEQ 0059

3000	003370	005737	001304		TST	\$PASS		;TEST IF FIRST PASS
3001	003374	001104			BNE	7\$;NO - SKIP TO DRIVE SELECT TEST
3002	003376	132737	000200	001317	BITB	#BIT7,\$ENVM		;TEST IF SHOULD SIZE
3003	003404	001402			BEQ	92\$;YES - SKIP TO DRIVE SIZING.
3004	003406	000137	003740		JMP	11\$		
3005	003412	005013			CLR	(R3)	92\$:	;CLEAR DEVICE MAP
3006	003414	123727	000041	000013	CMPB	2#41,#13		;TEST IF RK06 IS LOAD DEVICE
3007	003422	001066			BNE	77\$;NO - SKIP
3008	003424	022737	000001	001624	CMP	#1,\$RTFLG		;WAS START AT PARAM?
3009	003432	001406			BEQ	2\$;YES - SKIP
3010	003434	104401	050034		TYPE	,OPR007		;NO TEST OF DRIVE 0
3011	003440	042737	000001	001656	BIC	#DOTST,OPTFLG		;DR FLAG - NO TEST DRIVE 0
3012	003446	000457			BR	7\$		
3013	003450	104401	047664		TYPE	,OPR006	2\$:	;MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3014	003454	005037	001610		CLR	L.CS2		;SET TO DRIVE 0
3015	003460	005037	001232		CLR	\$TMP4		;CLEAR FOR USE AS A SWITCH
3016	003464	012737	000101	001600	MOV	#SELDRV,L.CS1	3\$:	;LOAD FOR SELECT
3017	003472	104417			TLOADRK			;LOAD RK & DO SELECT
3018								
3019	003474	104423			TWAT16			;WAIT 16MS FOR COMPLETION
3020	003476	104002			ERROR	2		;NOT DONE ON TIME
3021								
3022	003500	104420			TGETRK			;GET RK REGISTER
3023	003502	032737	100000	001540	BIT	#CERR,T.CS1		;TEST IF CERR
3024	003510	001414			BEQ	5\$;NO - SKIP
3025	003512	032737	010000	001550	BIT	#NED,T.CS2		;TEST IF NED
3026	003520	001002			BNE	4\$;YES - SKIP
3027								
3028	003522	104421			TCHKOP			;CHECK THE OPERATION AND REPORT THE ERROR
3029	003524	104004			ERROR	4 ;OR5,6,7		;AFTER THE ERROR IS REPORTED THE TEST
3030								;IS ABORTED
3031	003526	104401	050034		TYPE	,OPR007	4\$:	;TYPE NO TEST OF DRIVE 0
3032	003532	042737	000001	001656	BIC	#DOTST,OPTFLG		;DR FLAG - NO TEST OF DRIVE 0
3033	003540	000422			BR	7\$;SKIP OVER WAIT FOR PACK MOUNT
3034	003542	005737	001232		TST	\$TMP4	5\$:	;TEST FLAG DRIVE READY HAS RESET
3035	003546	001010			BNE	6\$;YES - SKIP TO CHECK IF IT IS SET AGAIN
3036	003550	032737	000200	001552	BIT	#DRDY,T.DS		;ELSE CHECK READY
3037	003556	001342			BNE	3\$;STILL SET - GET STATUS AGAIN
3038	003560	012737	177777	001232	MOV	#-1,\$TMP4		;ELSE SET FLAG TO INDICATE READY WENT LOW
3039	003566	000736			BR	3\$;GO GET STATUS AGAIN
3040								
3041	003570	032737	000200	001552	BIT	#DRDY,T.DS	6\$:	;TEST IF READY SET AGAIN
3042	003576	001732			BEQ	3\$;NO - GO GET STATUS AGAIN
3043	003600	052737	000001	001656	BIS	#DOTST,OPTFLG	77\$:	;ELSE SET DRV 0 TEST FLAG.
3044								
3045	003606	005000			CLR	R0	7\$:	;CLEAR FOR DRIVE NUMBER COUNTER
3046	003610	012701	000001		MOV	#1,R1		;SET BIT 0 AS DRIVE SELECTOR
3047								
3048	003614	032737	000001	001656	BIT	#DOTST,OPTFLG		;TEST DRIVE 0?
3049	003622	001430			BEQ	9\$;NO - SKIP
3050								
3051	003624	104416			TSSINIT		8\$:	;INITIALZE SUBSYSTEM
3052	003626	104003			ERROR	3		;ERROR IF NOT SUCCESSFUL
3053								
3054	003630	010037	001610		MOV	R0,L.CS2		;LOAD DRIVE NUMBER
3055	003634	012737	000101	001600	MOV	#SELDRV,L.CS1		;LOAD DRIVE SELECT


```

3056 003642 104417          TLOADRK          ;LOAD RK REGS
3057
3058 003644 104423          TWAT16          ;WAIT FOR INTERRUPT
3059 003646 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
3060
3061 003650 104420          TGETRK          ;GET RK REGS
3062 003652 032737 100000 001540  BIT #CERR,T.CS1 ;ERROR?
3063 003660 001017          BNE 10$        ;YES - SKIP
3064 003662 032737 000200 001552  BIT #DRDY,T.DS  ;ELSE TEST IF DRIVE READY
3065 003670 001405          BEQ 9$         ;NO - SKIP
3066 003672 032737 004000 001552  BIT #WRL,T.DS  ;ELSE TEST IF WRITE LOCKED
3067 003700 001001          BNE 9$         ;YES - SKIP
3068
3069 003702 050113          BIS R1,(R3) ;SET BIT - DRIVE PRESET IN MAP
3070
3071 003704 005200          9$: INC R0        ;BUMPS TO NEXT DRIVE
3072 003706 006301          ASL R1        ;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3073 003710 032701 000400          BIT #BIT8,R1  ;WAS LAST DRIVE DONE?
3074 003714 001743          BEQ 8$        ;YES - SKIP
3075 003716 000410          BR 11$       ;ELSE LOOP TO SELECT NEXT DRIVE
3076
3077 003720 032737 010000 001550 10$: BIT #NED,T.CS2 ;WAS CERR DUE TO NED?
3078 003726 001366          BNE 9$        ;YES - BUMP TO NEXT DRIVE
3079
3080 003730 104421          TCHKOP          ;ELSE REPORT THE ERRORS
3081 003732 104004          ERROR 4 ;OR5,6,7
3082 003734 000000          101$: .WORD 0
3083 003736 177777          100$: .WORD -1 ;A SWITCH - IT NEVER GETS EXECUTED
3084 003740 005737 003736          11$: TST 100$   ;TEST SWITCH
3085 003744 100036          BPL 16$       ;IF PLUS - SKIP DRIVE TEST MESSAGE
3086 003746 005237 003736          INC 100$     ;ELSE BUMP SWITCH TO PLUS
3087
3088 003752 005713          TST (R3)      ;ANY DRIVE AVAILABLE?
3089 003754 001004          BNE 12$      ;BR IF NOT ZERO
3090 003756 104401 050124          TYPE ,OPRO08 ;ELSE REPORT NO DRIVES AVAILABLE
3091 003762 000137 042442          JMP CTRHLT    ;GO TO CONTROLLED HALT
3092
3093 003766 012701 000200          12$: MOV #BIT7,R1 ;SET DRIVE SELECTOR FOR DRIVE 7
3094 003772 012700 000007          MOV #7,R0    ;SET DRIVE NUMBER TO 7
3095 003776 104401 050207          TYPE ,OPRO09 ;TYPE PREFIX TO DRIVE TEST MESSAGE
3096
3097 004002 030113          13$: BIT R1,(R3) ;TEST IF THIS DRIVE TO BE TESTED
3098 004004 001004          BNE 15$     ;YES - SKIP
3099
3100 004006 005300          14$: DEC R0    ;ELSE DECREMENT DRIVE NUMBER
3101 004010 006201          ASR R1      ;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3102 004012 001373          BNE 13$    ;IF NOT SHIFTED OUT - LOOP
3103 004014 000412          BR 16$     ;ELSE GO TO STATUS VALID TEST
3104
3105 004016 010037 003734          15$: MOV R0,101$ ;PUT DRIVE NUMBER IN TYPE LOCATION
3106 004022 052737 000060 003734  BIS #BIT4!BITS,101$ ;MAKE IT ASCIZ
3107 004030 104401          TYPE       ;TYPE IT
3108 004032 003734          101$
3109 004034 104401 047501          TYPE SPACE2 ;TYPE SOME SPACES
3110 004040 000762          BR 14$     ;LOOP
3111

```


K05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 62
T3 SELECT ALL DRIVES

SEQ 0062

```

3168 004254 032737 040000 001552 22$: BIT #DSC,T.DS ;TEST IF DSC RESET
3169 004262 001410 BEQ 23$ ;YES - SKIP
3170 004264 012737 055375 001470 MOV #EMDSC,EM12N
3171 004272 012737 055635 057702 MOV #EMSCLR,DF011A
3172 004300 104003 ERROR 3 ;"DSC NOT RESET RESULT OF SUBSYS CLEAR"
3173 004302 000723 BR 18$
3174
3175 004304 005737 001630 23$: TST DRVBIT ;TEST IF DRVBIT IS NEGATIVE
3176 004310 100320 BPL 18$ ;NO - SKIP
3177 004312 010137 001630 MOV R1,DRVBIT ;STORE DRIVE SELECT BIT
3178 004316 010037 001240 MOV R0,$TMP7 ;STORE DRIVE NUMBER TO BE TESTED
3179 004322 000713 BR 18$
3180
3181 004324 013737 001240 001626 21$: MOV $TMP7,DRVNUM ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
3182
3183 004332 023727 001624 000002 CMP SRTFLG,#2 ;TEST IF CLOCK ADJUST START
3184 004340 001002 BNE 25$ ;NO - SKIP
3185 004342 000137 042332 JMP ADJCLK ;GO TO ADJUST CLOCK ROUTINE
3186
3187 004346 25$:
3188
3189
3190 ;*****
3191 ;*TEST 4 RELEASE ALL DRIVES
3192 ;*
3193 ;* RELEASE ALL DRIVES. MAKE SURE NO ERROR
3194 ;* SETS AND STATUS VALID IS RESET.
3195 ;*
3196 ;*****
3196 004346 000004 TST4: SCOPE
3197 004350 012737 000062 001262 MOV #50,$TIMES ;DO 50. ITERATIONS
3198 004356 104416 TSSINIT ;INITIALIZE SUBSYSTEM
3199 004360 104003 ERROR 3 ;BAD INIT
3200
3201 004362 013737 001626 001610 MOV DRVNUM,L.CS2 ;SET DRIVE NUMBER
3202 004370 012737 000101 001600 MOV #SELDRV,L.CS1 ;SET DRIVE SELECT
3203
3204 004376 104417 TLOADRK ;LOAD RK REGS
3205 004400 104423 TWAT16 ;WAIT FOR INTERRUPT
3206 004402 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3207
3208 004404 104421 TCHKOP ;CHECK FOR ANY ERRORS
3209 004406 104004 ERROR 4 ;OR5,6,7 ;REPORT ANY ERRORS
3210
3211 004410 012737 000010 001610 MOV #RLS,L.CS2 ;SET DRIVE RELEASE,STILL SET FOR SELECT
3212
3213 004416 104417 TLOADRK ;LOAD RK REGS
3214 004420 104423 TWAT16 ;WAIT FOR INTERRUPT
3215 004422 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3216
3217 004424 104421 TCHKOP ;CHECK FOR ANY ERRORS
3218 004426 104004 ERROR 4 ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
3219 004430 032737 100000 001552 BIT #SVAL,T.DS ;DID SVAL RESET?
3220 004436 001404 BEQ 1$ ;YES - SKIP
3221 004440 012737 054500 001400 MOV #EM49,EM3N ;MESSAGE (SVAL NOT RESET W/RELEASE)
3222 004446 104003 ERROR 3
3223 004450 1$:

```


3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279

004450

004450	000004		
004452	012737	000062	001262
004460	104416		
004462	104003		
004464	012701	000001	
004470	013737	001626	001610
004476	012737	000101	001600
004504	005037	001616	
004510	104417		
004512	104423		
004514	104002		
004516	104421		
004520	104004		
004522	032737	100000	001552
004530	001007		
004532	012737	055235	001460
004540	012737	046676	057702
004546	104011		
004550	010137	001616	
004554	104417		
004556	104423		
004560	104002		
004562	104421		
004564	104004		
004566	032737	100000	001552
004574	001407		
004576	012737	055235	001470
004604	012737	055252	057702
004612	104012		
004614	022701	000003	
004620	001402		
004622	005201		
004624	000727		
004626			

TSTLUP:

```

*****
*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.
*****

```

TST5:

```

SCOPE
MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
TSSINIT      ;;CLEAR SUBSYSTEM
ERROR      3      ;;BAD CLEAR MESSAGE
MOV      #1,R1      ;;PRESET R1 FOR MESSAGE PAIR 1
MOV      DRVNUM,L.CS2      ;;LOAD DRV NUMBER
MOV      #SELD, L.CS1      ;;LOAD SELECT COMMAND
1$:      CLR      L.MR1      ;;LOAD FOR STANDARD STATUS
TLOADRK      ;;LOAD RK
TWTAT16      ;;WAIT FOR INTERRUPT
ERROR      2      ;;TO SLOW/NOT COMPLETE ERROR
TCHKOP      ;;CHECK OPERATION
ERROR      4 ;5,6 OR 7      ;;REPORT ALL ERRORS

BIT      #SVAL,T.DS      ;;TEST STATUS VALID SET
BNE      2$      ;;YES-SKIP

MOV      #EMSVAL,EM11N
MOV      #EMSELD,DF011A
ERROR      11      ;"SVAL NOT SET RESULT OF DRIVE SELECT"

2$:      MOV      R1,L.MR1      ;;LOAD MESSAGE PAIR SELECT

TLOADRK      ;;LOAD RK
TWTAT16      ;;WAIT FOR INTERRUPT
ERROR      2      ;;TO SLOW/NOT COMPLETE ERROR

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

BIT      #SVAL,T.DS      ;;TEST STATUS VALID RESET
BEQ      3$      ;;YES-SKIP

MOV      #EMSVAL,EM12N
MOV      #EMNZPR,DF011A
ERROR      12      ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"

3$:      CMP      #3,R1      ;;WAS PAIR 3 SELECTED?
BEQ      4$      ;;YES-SKIP
INC      R1      ;;BUMP TO NEXT PAIR
BR      1$      ;;SKIP TO DO IT.

```

4\$:

```

*****
*TEST 6      WRITING CS2 AND STATUS VALID

```


M05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.F11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 64
T6 WRITING CS2 AND STATUS VALID

SEQ 0064

```

3280
3281
3282
3283
3284
3285
3286 004626 000004
3287 004630 012737 000062 001262
3288 004636 104416
3289 004640 104003
3290
3291 004642 013737 001626 001610
3292 004650 012737 000101 001600
3293
3294 004656 104417
3295 004660 104423
3296 004662 104002
3297
3298 004664 104421
3299 004666 104004
3300
3301 004670 032737 100000 001552
3302 004676 001007
3303
3304 004700 012737 055235 001460
3305 004706 012737 046676 057702
3306 004714 104011
3307
3308 004716 013762 001626 000010 1$:
3309
3310 004724 104420
3311
3312 004726 032737 100000 001552
3313 004734 001407
3314
3315 004736 012737 055235 001470
3316 004744 012737 055313 057702
3317 004752 104012
3318 004754 2$:
3319

```

```

;*
;* SELECT AN AVAILABLE DRIVE. MAKE SURE STATUS
;* VALID IS SET. WRITE COMMAND AND STATUS REGISTER 2.
;* MAKE SURE STATUS VALID RESETS.
;*
;*****
1ST6: SCOPE
MOV #50.,$TIMES ;:DO 50. ITERATIONS
TSSINIT ;:CLEAR SUBSYSTEM
ERROR 3 ;:BAD INIT ERROR

MOV DRVNUM,L.CS2 ;:LOAD DRIVE NUMBER
MOV #SELD, L.CS1 ;:LOAD DRIVE SELECT

TLOADRK ;:LOAD RK
TWTAT16 ;:WAIT FOR INTERRUPT
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR

TCHKOP ;:CHECK OPERATION
ERROR 4 ;5,6, OR 7 ;:REPORT ALL ERRORS

BIT #SVAL,T.DS ;:TEST STATUS VALID SET
BNE 1$ ;:YES-SKIP

MOV #EMSVAL,EM11N
MOV #EMSELD,DF011A
ERROR 11 ;:"SVAL NOT SET RESULT OF DRV SELECT"

MOV DRVNUM,RKCS2(R2) ;:WRITE CS2 TO RESET SVAL

TGETRK ;:GET RK REGS.

BIT #SVAL,T.DS ;:TEST SVAL RESET
BEQ 2$ ;:YES-SKIP

MOV #EMSVAL,EM12N
MOV #EMWCS2,DF011A
ERROR 12 ;:"SVAL NOT RESET BY WRITING CS2"

```


.SBTTL **CONTROLLER ERROR TESTS

3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375

004754 000004
004756 012737 000062 001262
004764 104416
004766 104003
004770 013737 001626 001610
004776 012737 000101 001600
005004 052737 002000 001600
005012 104417
005014 104423
005016 104002
005020 104422
005022 000040
005024 000000
005026 000000
005030 104004
005032 032737 100000 001552
005040 001007
005042 012737 055235 001460
005050 012737 053751 057702
005056 104011
005060
005062 000004
005070 012737 000062 001262
005070 104416

```
*****  
*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 #SELDLV,L.CS1 ;:LOAD DRIVE SELECT  
BIS #CDT,L.CS1 ;:LOAD DRIVE TYPE  
  
TLOADRK ;:LOAD RK  
TWTAT16 ;: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,EMI1N  
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
```


3376	005072	104003			ERROR 3		:BAD INIT ERROR
3377							
3378	005074	013737	001626	001610	MOV	DRVNUM,L.CS2	:LOAD DRIVE NUMBER
3379	005102	012737	000101	001600	MOV	#SELDIV,L.CS1	:LOAD DRIVE SELECT
3380	005110	012737	000020	001616	MOV	#PAT,L.MR1	:LOAD EVEN PARITY BIT
3381							
3382	005116	104417			TLOADRK		:LOAD RK REGS-SELECT W/EVEN PARITY
3383	005120	104423			TWAIT6		:WAIT FOR INTERRUPT
3384	005122	104002			ERROR 2		:TO SLOW/NOT COMPLETE ERROR
3385							
3386	005124	104422			TCHKWE		:CHECK OPERATION FOR EXPECTED ERROR
3387	005126	000011			DRPARERR:SPARERR		:DRIVE SELECTED DRIVE BUS PARITY ERROR
3388	005130	000000			.WORD 0		:CONTROLLER DETECTED DRIVE BUS PARITY ERROR
3389	005132	000000			.WORD 0		
3390	005134	104004			ERROR 4 ; OR 5,6,7		:REPORT ANY DIFFERENCES
3391							
3392	005136	012700	000400		MOV	#BITS,RO	:ROUTINE TO DETERMINE WHICH BIT
3393							
3394	005142	013701	001626		MOV	DRVNUM,R1	:SHOULD BE SET IN ASOF TO INDICATE
3395	005146	001403			BEQ	3\$:DRIVE ATTENTION. RO WILL HAVE THE
3396	005150	006300		2\$:	ASL	RO	:BIT THAT SHOULD BE SET FOR THE DRIVE
3397	005152	005301			DEC	R1	:IN USE
3398	005154	001375			BNE	2\$	
3399							
3400	005156	030037	001556	3\$:	BIT	RO,T.ASOF	:TEST ATTENTION SET
3401	005162	001007			BNE	4\$:YES-SKIP
3402	005164	012737	055421	001460	MOV	#EMDA,EM11N	
3403	005172	012737	053666	057702	MOV	#EMDPAR,DF011A	
3404	005200	104011			ERROR	11	: "DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3405	005202	032737	040000	001540	4\$:	BIT	#DI,T.CS1
3406	005210	001007			BNE	5\$:TEST DRIVE INTERRUPT SET
3407	005212	012737	055355	001460	MOV	#EMDI,EM11N	:YES-SKIP
3408	005220	012737	053666	057702	MOV	#EMDPAR,DF011A	
3409	005226	104011			ERROR	11	: "DRV INT NOT SET RESULT OF DRV PARITY ERR"
3410							
3411	005230	032737	040000	001552	5\$:	BIT	#DSC,T.DS
3412	005236	001007			BNE	6\$:TEST DRIVE STATUS CHANGE SET
3413	005240	012737	055375	001460	MOV	#EMDSC,EM11N	:YES-SKIP
3414	005246	012737	053666	057702	MOV	#EMDPAR,DF011A	
3415	005254	104011			ERROR	11	: "DSC NOT SET RESULT OF DRV PARITY ERR"
3416							
3417	005256	032737	100000	001552	6\$:	BIT	#SVAL,T.DS
3418	005264	001007			BNE	7\$:TEST STATUS VALID SET
3419	005266	012737	055235	001460	MOV	#EMSVAL,EM11N	:YES-SKIP
3420	005274	012737	053666	057702	MOV	#EMDPAR,DF011A	
3421							
3422	005302	104011			ERROR	11	: "SVAL NOT SET RESULT OF DRV PAR ERR"
3423							
3424	005304	005037	001616		7\$:	CLR	L.MR1
3425							:CLEAR PAT IN MR1
3426	005310	052737	100000	001600	BIS	#CCLR,L.CS1	:CLEAR CONTROLLER
3427	005316	104417			TLOADRK		:LOAD RK REGS TO DO CLEAR
3428							
3429							
3430	005320	104421			TCHKOP		:CHECK NO ERRORS SET
3431	005322	104004			ERROR 4 ; OR 5,6,7		:REPORT ALL ERRORS STILL SET


```

3488 005574 010137 001616      14$:  MOV      R1,L.MR1      ;LOAD STATUS PAIR SELECTION
3489 005600 104417                TLOADRK      ;LOAD RK REGS
3490 005602 104423                TWAT16       ;WAIT FOR INTERRUPT
3491 005604 104002                ERROR      2   ;TO SLOW/NOT COMPLETE ERROR
3492 005606 104421                TCHKOP      ;CHECK IF ANY ERRORS SET
3493 005610 104004                ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS SET.
3494 005612 030037 001556                BIT      RD,T.ASOF    ;TEST ATTENTION STILL SET
3495 005616 001007                BNE      15$        ;YES-SKIP
3496 005620 012737 055421 001410                MOV      #EMDA,EM4N
3497 005626 012737 055252 057702                MOV      #EMNZPR,DF011A
3498 005634 104014                ERROR      14       ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
3501 005636 032737 040000 001540      15$:  BIT      #DI,T.CS1
3502 005644 001007                BNE      16$
3503 005646 012737 055355 001510                MOV      #EMDI,EM14N
3504 005654 012737 055252 057702                MOV      #EMNZPR,DF011A
3505 005662 104014                ERROR      14       ;"DRV INT RESET RESULT OF NON-0 PAIP SELECT"
3507 005664 005201                16$:  INC      R1          ;BUMP PAIR SELECT
3508 005666 022701 000004                CMP      #4,R1      ;ALL PAIRS DONE?
3509 005672 001340                BNE      14$        ;NO-LOOP
3511 005674 005037 001616                CLR      L.MR1      ;CLEAR MR1
3512 005700 012737 000105 001600                MOV      #CLEAR,L.CS1 ;LOAD DRIVE CLEAR
3513 005706 104417                TLOADRK      ;DO DRIVE CLEAR
3514 005710 104423                TWAT16       ;WAIT FOR INTERRUPT
3515 005712 104002                ERROR      2   ;TO SLOW/NOT COMPLETE ERROR
3516 005714 104421                TCHKOP      ;CHECK FOR ANY ERRORS
3517 005716 104004                ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS
3518 005720 012701 000020                17$:  MOV      #20,R1     ;SET COUNT FOR SHORT WAIT
3519 005724 005301                DEC      R1         ;TO ALLOW CONTROLLER TIME TO POLL
3520 005726 001376                BNE      17$        ;DRIVES
3521 005730 104420                TGETRK      ;GET RK REGS
3522 005732 030037 001556                BIT      RD,T.ASOF    ;TEST ATTENTION RESET
3523 005736 001407                BEQ      18$        ;YES-SKIP
3524 005740 012737 055421 001470                MOV      #EMDA,EM12N
3525 005746 012737 046724 057702                MOV      #EMDCLR,DF011A
3526 005754 104012                ERROR      12       ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
3527 005756 032737 040000 001540      18$:  BIT      #DI,T.CS1    ;TEST DRIVE INTERRUPT RESET
3528 005764 001407                BEQ      19$        ;YES-SKIP
3529 005766 012737 055355 001470                MOV      #EMDI,EM12N
3530 005774 012737 046724 057702                MOV      #EMDCLR,DF011A
3531 006002 104012                ERROR      12       ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
3532 006004                19$:  *****
3533 006004                ;*TEST 11      UNIT FIELD ERROR ON RELEASE
3534 006004                ;*

```


E06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.F11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 69
T11 UNIT FIELD ERROR ON RELEASE

SEQ 0069

3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599

006004 000004
006006 012737 000062 001262
006014 104416
006016 104002
006020 013737 001626 001610
006026 012737 000101 001600
006034 104417
006036 104423
006040 104002
006042 104421
006044 104004
006046 052737 000010 001610
006054 012737 000040 001616
006062 104417
006064 004437 035050
006070 023
006071 002
006072 042762 000040 000026
006100 104423
006102 104002
006104 104422
006106 000000
006110 000000
006112 000004
006114 104004
006116 104416
006120 104002
006122 000004

```

: *      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.
: *
: *****
: ST11:  SCOPE
:        MOV      #50.,$TIMES      ;:DO 50. ITERATIONS
:        TSSINIT          ;:CLEAR SUBSYSTEM
:        ERROR  2          ;:BAD INIT ERROR
:
:        MOV      DRVNUM,L.CS2     ;:SELECT A DRIVE
:        MOV      #SELDRV,L.CS1    ;:DO DRIVE SELECT
:
:        TLOADRK          ;:LOAD RK
:        TWAT16          ;:WAIT FOR INTERRUPT
:        ERROR  2          ;:TO SLOW/NOT COMPLETE ERROR
:
:        TCHKOP          ;:CHECK FOR ANY ERRORS
:        ERROR  4 ; OR 5,6,7      ;:REPORT ALL ERRORS.
:
:        BIS      #RLS,L.CS2       ;:LOAD RELEASE
:        MOV      #DMD,L.MR1       ;:SET DIAGNOSTIC MODE
:
:        TLOADRK          ;:LOAD RK
:
:        JSR      R4,MCLOCK        ;:CALL MAINT CLOCK
:        .BYTE  1D19              ;:NUMBER OF PHASES
:        .BYTE  2                  ;:NUMBER OF CLOCK XISTIONS
:
:        BIC      #DMD,RKMR1(R2)   ;:CLEAR DIAG MODE
:
:        TWAT16          ;:WAIT FOR INTERRUPT
:        ERROR  2          ;:TO SLOW/NOT COMPLETED
:
:        TCHKWE          ;:CHECK OPERATION WITH ERROR
:        .WORD  0
:        .WORD  0
:        .WORD  UFERR             ;:UNIT FIELD ERROR
:        ERROR  4 ; OR 5,6,7      ;:REPORT ANY DISCREPENCIES
:
:        TSSINIT          ;:CLEAR SUBSYSTEM TO INSURE UFE RESETS
:        ERROR  2
: *****
: TEST 12  UNIT FIELD ERROR ON SELECT
:
:      ISSUE A SUBSYSTEM CLEAR.  SELECT AN AVAILABLE
:      DRIVE.  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.
: *****
: ST12:  SCOPE

```


F06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 70
T12 UNIT FIELD ERROR ON SELECT

SEQ 0070

3600	006124	012737	000062	001262	MOV	#50.,\$TIMES	::DO 50. ITERATIONS
3601	006132	104416			TSSINIT		:CLEAR SUBSYSTEM
3602	006134	104003			ERROR	3	:BAD INIT ERROR
3603							
3604	006136	013737	001626	001610	MOV	DRVNUM,L.CS2	:LOAD DRIVE NUMBER
3605	006144	012737	000101	001600	MOV	#SELDIV,L.CS1	:LOAD DRIVE SELECT
3606							
3607	006152	104417			TLOADRK		:LOAD RK
3608	006154	104423			TWAT16		:WAIT FOR INTERRUPT
3609	006156	104002			ERROR	2	:TO SLOW/NOT COMPLETE
3610							
3611	006160	104421			TCHKOP		:CHECK FOR ANY ERROR
3612	006162	104004			ERROR	4 ; OR 5,6,7	:REPORT ALL ERRORS
3613							
3614	006164	012737	000043	001616	MOV	#DMD!BIT1!BIT0,L.MR1	:LOAD DIAG MODE & MSG PAIR 3
3615	006172	005037	001610		CLR	L.CS2	:LOAD FOR DRIVE 0
3616							
3617	006176	104417			TLOADRK		:LOAD RK
3618							
3619	006200	004437	035050		JSR	R4,MCLOCK	:CALL MAINTENANCE CLOCK
3620	006204	026			.BYTE	↑D22	:THROUGH PHASE 6
3621	006205	002			.BYTE	2	:PLUS 2 TRANSITIONS
3622							
3623	006206	042762	000040	000026	BIC	#DMD,RKMR1(R2)	:CLEAR DIAG MODE
3624							
3625	006214	104423			TWAT16		:WAIT FOR INTERRUPT
3626	006216	104002			ERROR	2	:TO SLOW/NOT COMPLETED ERROR
3627							
3628	006220	104422			TCHKWE		:CHECK OPERATION WITH ERROR
3629	006222	000000			.WORD	0	
3630	006224	000000			.WORD	0	
3631	006226	000004			.WORD	UFERR	:UNIT FIELD ERROR SHOULD SET
3632	006230	104004			ERROR	4 ; OR 5,6,7	:REPORT ANY DISCREPENCIES

.SBTTL **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)

3648					TST13:	SCOPE	
3649	006232	000004			MOV	#50.,\$TIMES	::DO 50. ITERATIONS
3650	006234	012737	000062	001262	TSSINIT		:CLEAR SUBSYSTEM
3651	006242	104416			ERROR	3	:BAD INIT ERROR
3652	006244	104003					
3653							
3654	006246	013737	001626	001610	MOV	DRVNUM,L.CS2	:LOAD DRIVE NUMBER
3655	006254	012737	000113	001600	MOV	#RECAL,L.CS1	:LOAD RECAL


```

3656
3657 006262 104417          TLOADRK          ;LOAD RK
3658 006264 104423          TWAT16          ;WAIT FOR 1ST INTERRUPT
3659 006266 104002          ERROR 2        ;TO SLOW/NOT COMPLETE
3660 006270 005037 001662    CLR INTSET      ;CLEAR INTERRUPT FLAG
3661 006274 104420          TGETRK          ;GET RK REGS
3662 006276 032737 100000 001552 BIT #SVAL,T.DS  ;TEST SVAL SET
3663 006304 001010          BNE 1$         ;YES-SKIP
3664 006306 012737 055235 001460 MOV #EMSVAL,EM11N
3665 006314 012737 046765 057702 MOV #EMRCAL,DF011A
3666 006322 104011          ERROR 11       ;"SVAL NOT SET RESULT OF RECAL"
3667 006324 000463          BR 50$        ;ABORT TEST
3668
3669 006326 104437          1$: TWAT8S      ;WAIT FOR INTERRUPT
3670 006330 000401          BR 2$         ;NO INTERRUPT RETURN
3671 006332 000404          BR 3$         ;INTERRUPT RETURN
3672
3673 006334 012737 054550 001370 2$: MOV #EM50,EM2N ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
3674 006342 104002          ERROR 2
3675
3676 006344 104420          3$: TGETRK      ;GET RK REGS
3677 006346 032737 100000 001552 BIT #SVAL,T.DS  ;TEST SVAC SET NOW
3678 006354 001410          BEQ 4$        ;NO-SKIP
3679 006356 012737 055235 001470 MOV #EMSVAL,EM12N
3680 006364 012737 055462 057702 MOV #EM2INT,DF011A
3681 006372 104012          ERROR 12       ;"SVAL NOT RESET RESULT OF SECOND TEST"
3682 006374 000437          BR 50$
3683
3684 006376 032737 040000 001552 4$: BIT #DSC,T.DS  ;TEST DSC SET BY ATTENTION
3685 006404 001010          BNE 5$        ;YES-SKIP
3686 006406 012737 055375 001460 MOV #EMDSC,EM11N
3687 006414 012737 055462 057702 MOV #EM2INT,DF011A
3688 006422 104011          ERROR 11       ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
3689 006424 000423          BR 50$
3690
3691 006426 012737 000101 001600 5$: MOV #SELDRV,L.CS1 ;LOAD DRIVE SELECT
3692
3693 006434 104417          TLOADRK          ;LOAD RK REGS
3694 006436 104423          TWAT16          ;WAIT FOR INTERRUPT
3695 006440 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
3696
3697 006442 104421          TCHKOP          ;CHECK FOR ANY ERRORS
3698
3699 006444 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
3700
3701 006446 032737 100000 001552 BIT #SVAL,T.DS  ;TEST SVAC SET
3702 006454 001007          BNE 50$       ;YES-SKIP
3703 006456 012737 055235 001460 MOV #EMSVAL,EM11N
3704 006464 012737 046676 057702 MOV #EMSELD,DF011A
3705 006472 104011          ERROR 11       ;"SVAL NOT SET RESULT OF DRV SEL.
3706 006474
3707 *****
3708 *TEST 14 SINGLE INTERRUPT FROM ATTENTION
3709 *
3710 * DO A SEEK TO CYLINDER 0. WAIT FOR INTERRUPT FROM
3711 * DRIVE ATTENTION. LOWER PRIORITY AGAIN AND MAKE

```


H06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 72
T14 SINGLE INTERRUPT FROM ATTENTION

SEQ 0072

```
3712          ;* SURE ANOTHER INTERRUPT DOES NOT OCCUR. CLEAR DRIVE.
3713          ;*
3714          ;*****
3715 006474 000004          †ST14: SCOPE
3716 006476 012737 000062 001262  MOV #50.,$TIMES ;:DO 50. ITERATIONS
3717 006504 104416          TSSINIT ;:CLEAR SUBSYSTEM
3718 006506 104003          ERROR 3 ;:BAD INIT ERROR
3719
3720 006510 013737 001626 001610  MOV DRVNUM,L.CS2 ;:LOAD DRIVE NUMBER
3721 006516 012737 000117 001600  MOV #SEEK,L.CS1 ;:LOAD SEEK DCYL LEFT AT 0.
3722
3723 006524 104417          TLOADRK ;:LOAD RK REGS
3724 006526 104423          TWAT16 ;:WAIT FOR INTERRUPT
3725 006530 104002          ERROR 2 ;:TO SLOW/NOT COMPLETED ERROR
3726
3727
3728 006532 104420          TGETRK ;:GET RK REGS
3729
3730 006534 032737 040000 001540  BIT #DI,T.CS1 ;:TEST DI SET
3731 006542 001010          BNE 2$ ;:YES-SKIP
3732 006544 012737 055355 001460  MOV #EMDI,EM11N
3733 006552 012737 055503 057702  MOV #EMSKSF,DF011A
3734 006560 104011          ERROR 11 ;:"DI NOT SET RESULT OF SEEK TO SELF"
3735 006562 000417          BR 50$
3736
3737
3738 006564 012700 000031          2$: MOV #25.,R0 ;:LOAD AND DECREMENT A COUNT TO
3739 006570 005300          3$: DEC R0 ;:ZERO. GIVE CONTROLLER A CHANCE TO
3740 006572 001376          BNE 3$ ;:INTERRUPT AGAIN. ERROR IF IT DOES.
3741
3742 006574 022737 000001 001662  CMP #1,INTSET ;:CHECK ONLY ONE INTERRUPT OCCURRED
3743 006602 001407          BEQ 50$ ;:YES-SKIP
3744 006604 012737 055655 001500  MOV #EMMI,EM13N
3745 006612 012737 055503 057702  MOV #EMSKSF,DF011A
3746 006620 104013          ERROR 13 ;:"MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"
3747
3748 006622 104421          50$: TCHKOP ;:CHECK FOR ANY ERRORS
3749 006624 104004          ERROR 4 ;OR 5,6,7 ;:REPORT ALL ERRORS
3750          ;*****
3751          ;*TEST 15 RESET ATTENTIONS WITH UNIBUS INIT
3752          ;*
3753          ;* DO A SEEK TO CYLINDER 0 ON ALL AVAILIABLE DRIVES.
3754          ;* ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.
3755          ;*
3756          ;*****
3757 006626 000004          †ST15: SCOPE
3758 006630 012737 000012 001262  MOV #10.,$TIMES ;:DO 10. ITERATIONS
3759 006636 104416          TSSINIT ;:CLEAR SUBSYSTEM
3760 006640 104003          ERROR 3 ;:BAD INIT ERROR
3761
3762 006642 013737 001626 001610  MOV DRVNUM,L.CS2 ;:LOAD DRIVE NUMBER
3763 006650 012737 000117 001600  MOV #SEEK,L.CS1 ;:LOAD SEEK (TO SELF-0)
3764
3765 006656 104417          TLOADRK ;:LOAD RK REGS
3766 006660 104423          TWAT16 ;:WAIT FOR INTERRUPT
3767 006662 104002          ERROR 2 ;:TO SLOW/NOT COMPLETE
```



```

3768
3769 006664 104420          TGETRK          ;GET RK REGS
3770
3771 006666 032737 040000 001540 BIT      #DI,T.CS1    ;TEST DI SET
3772 006674 001010          BNE      1$      ;YES-EXIT
3773 006676 012737 055355 001460 MOV      #EMDI,EM11N
3774 006704 012737 055503 057702 MOV      #EMSKSF,DF011A
3775 006712 104011          ERROR     11      ;"DI NOT SET RESULT OF SEEK TO SELF
3776 006714 000450          BR        50$
3777
3778 006716 005037 001662          1$: CLR      INTSET    ;CLEAR INTERRUPT COUNTER
3779 006722 000005          RESET     ;DO UNIBUS RESET
3780 006724 004737 043664          JSR      PC,$TKINT ;RESET KEYBOARD INTERRUPT
3781
3782 006730 005037 001660          CLR      LCLKTK   ;CLEAR TICK COUNTER
3783 006734 004737 034214          JSR      PC,MYTIME ;CALL TIMER
3784 006740 022737 000012 001660 5$: CMP      #10.,LCLKTK ;COUNT 10 TICKS (MILLISECONDS)?
3785 006746 001372          BNE      5$      ;NO - LOOP
3786
3787 006750 012762 000100 000000 MOV      #IE,RKCS1(R2) ;SET IE FOR ANY STRAY INTERRUPTS
3788 006756 004737 033704          JSR      PC,$PTTST ;SET UP OPTIONS AGAIN
3789
3790 006762 104423          TWAT16
3791 006764 000410          BR        2$      ;WAIT 16 MS FOR AN INTERRUPT
3792                                     ;NONE IS EXPECTED SO RETURN SHOULD BE
3793 006766 012737 055520 001500 MOV      #EMUXIT,EM13N ;HERE-BR TO CONTINUE TEST.
3794 006774 012737 055562 057702 MOV      #EMRSET,DF011A ;INT OCCURRED ON RESET
3795 007002 104013          ERROR     13      ;"UNEXECUTED INTERRUPT RESULT OF RESET"
3796 007004 000414          BR        50$
3797 007006 104420          2$: TGETRK          ;GET RK REGS
3798 007010 032737 040000 001540 BIT      #DI,T.CS1    ;TEST DI RESET
3799 007016 001407          BEQ      50$     ;YES-SKIP
3800 007020 012737 055355 001470 MOV      #EMDI,EM12N
3801 007026 012737 055562 057702 MOV      #EMRSET,DF011A
3802 007034 104012          ERROR     12      ;"DI NOT RESET RESULT OF RESET"
3803
3804 007036          50$:
3805
3806 .SBTTL **ILLEGAL DISK ADDRESS ERROR TESTS
3807
3808 ;:*****
3809 ;:TEST 16      ILLEGAL DISK ADDRESS (PART 1)
3810 ;:
3811 ;:      ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
3812 ;:      ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
3813 ;:      CLEAR CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7,
3814 ;:      CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
3815 ;:      HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
3816 ;:
3817 ;:*****
3818 007036 000004          TST16: SCOPE
3819 007040 012737 000062 001262 MOV      #50,$TIMES ;:DO 50. ITERATIONS
3820 007046 012701 000003          MOV      #3,R1    ;:PRESET FOR SELECTING TRACK 3
3821
3822 007052 104416          TSSINIT          ;CLEAR SUBSYSTEM
3823 007054 104003          ERROR     3

```


JOB

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 74
T16 ILLEGAL DISK ADDRESS (PART 1)

SEQ 0074

```

3824
3825 007056 012737 000113 001600      MOV      #RECAL,L.CS1      ;SET UP TO RECAL
3826 007064 013737 001626 001610      MOV      DRVNUM,L.CS2     ;LOAD DRIVE
3827
3828 007072 104417                      TLOADRK                      ;LOAD RK REGS
3829
3830 007074 104423                      TWAT16                       ;WAIT FOR 1ST INTERRUPT
3831 007076 104002                      ERROR 2                      ;TO SLOW/NOT COMPLETE ERROR
3832
3833 007100 005037 001662                      CLR      INTSET             ;CLEAR INTERRUPT FLAG
3834
3835 007104 104437                      TWAT8S                       ;WAIT FOR INTERRUPT
3836 007106 104002                      ERROR 2
3837
3838 007110 012737 007116 001110      MOV      #1$, $LPERR        ;SET LOCAL LOOP ON ERROR
3839
3840 007116 104416                      1$: TSSINIT                   ;CLEAR SUBSYSTEM
3841 007120 104003                      ERROR 3                      ;BAD INIT ERROR
3842
3843 007122 013737 001626 001610      MOV      DRVNUM,L.CS2     ;LOAD DRIVE NUMBER
3844 007130 012737 000117 001600      MOV      #SEEK,L.CS1      ;LOAD SEEK
3845 007136 110137 001607                      MOVB     R1,L.DT           ;LOAD TRACK
3846
3847 007142 104417                      TLOADRK                      ;LOAD RK REGS
3848 007144 104423                      TWAT16                       ;WAIT FOR INTERRUPT
3849 007146 104002                      ERROR 2                      ;TO SLOW/NOT COMPLETE
3850
3851 007150 032701 000001                      BIT      #BIT0,R1          ;TEST IF HEAD ADDRESS HAS BIT 0
3852 007154 001403                      BEQ     2$                  ;NO - SKIP
3853 007156 032701 000002                      BIT      #BIT1,R1          ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3854 007162 001007                      BNE     3$                  ;YES-GO CHECK BOTH IDAE AND SKI SET
3855
3856 007164 104422                      2$: TCHKWE                     ;CHECK OPERATION WITH ERROR
3857 007166 002000                      IDAERR                      ;ILLEGAL DISK ADDRESS ERROR
3858 007170 000000                      0
3859 007172 000000                      0
3860 007174 104004                      ERROR 4 ; OR 5,6,7         ;REPORT ALL DISCREPANCIES
3861 007176 104415                      SCOP1                       ;LOCAL LOOP ON ERROR
3862 007200 000406                      BR      4$
3863
3864 007202 104422                      3$: TCHKWE                     ;CHECK OPERATION WITH ERROR
3865 007204 002002                      IDAERR!SKIERR              ;ILLEGAL DISK ADDRESS ERROR
3866 007206 000000                      0                            ;SEEK INCOMPLETE
3867 007210 000000                      0
3868 007212 104004                      ERROR 4 ;OR 5,6,7         ;REPORT ANY DISCREPANCIES
3869 007214 104415                      SCOP1                       ;LOCAL LOOP ON ERROR TO 1$
3870
3871 007216 005201                      4$: INC      R1              ;ELSE BUMP TO NEXT ILLEGAL TRACK
3872 007220 022701 000010      CMP      #8.,R1            ;ALL ILLEGAL TRACKS SELECTED?
3873 007224 001334                      BNE     1$                  ;NO-LOOP
3874
3875 ;*****
3876 ;*TEST 17      ILLEGAL DISK ADDRESS (PART 2)
3877 ;*
3878 ;*      ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
3879 ;*      ILLEGAL DISK ADDRESS ERROR SETS.  CLEAR CONTROLLER AND DRIVE
3879 ;*

```



```

3880
3881 007226 000004
3882 007230 012737 000062 001262
3883 007236 104416
3884 007240 104003
3885
3886 007242 012737 000113 001600
3887 007250 013737 001626 001610
3888
3889 007256 104417
3890
3891 007260 104423
3892 007262 104002
3893
3894 007264 005037 001662
3895
3896 007270 104437
3897 007272 104002
3898
3899 007274 012737 007302 001110
3900
3901 007302 104416
3902 007304 104003
3903
3904 007306 013737 001626 001610
3905 007314 012737 000117 001600
3906 007322 012737 001000 001614
3907
3908 007330 104417
3909 007332 104423
3910 007334 104002
3911
3912 007336 104422
3913 007340 002000
3914 007342 000000
3915 007344 000000
3916 007346 104004
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935

```

```

*****
TST17: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

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$, $LPERR ;SET LOOP TO BYPASS RECAL

1$: TSSINIT ;CLEAR SUBSYSTEM
ERROR 3

MOV DRVNUM, L.CS2 ;LOAD DRIVE NUMBER
MOV #SEEK,L.CS1 ;LOAD SEEK
MOV #1000, L.DCYL ;LOAD ILLEGAL CYLINDER

TLOADRK ;LOAD RK REGS
TWAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

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
*****
*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.
*
* 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

```


L06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 76
T20 READ BAD SECTOR INFORMATION

SEQ 0076

```

3936          ;*          SECTORS 24 SECTOR MODE. IF THE NUMBER OF BAD SECTORS FOR
3937          ;*          24 OR 26 SECTOR MODE EXCEED 20(10) THE DRIVE IS REMOVED
3938          ;*          FROM TESTING.
3939          ;*
3940          ;*          NOTE: THIS TEST IS RUN IN THE FIRST (QUICK
3941          ;*          VERIFY) PASS ONLY.
3942          ;*
3943          ;*****
3944          TST20: SCOPE
3945          MOV      #1,$TIMES          ;;DO 1 ITERATION
3946          CLRB    2$                ;CLEAR SECTOR POINTER
3947          TST     $PASS              ;TEST IF FIRST PASS
3948          BEQ     15$                ;NO - SKIP
3949          JMP     28$                ;ELSE EXIT TEST
3950          15$:
3951          CLR     R0
3952          CLR     R5                ;CLEAR R5 FOR BAD SECTOR COUNTING
3953          MOV     BSF26P,R3          ;SET POINT IN TO STORE BS 26 SECT FORMAT
3954          MOV     #1$,$LPERR         ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
3955          TSSINIT
3956          ERROR   3                ;BAD INIT ERROR
3957
3958          JSR     R4,LRLOAD          ;LOAD "L" REGS WITH
3959          RDDATA          ; READ DATA
3960          -400          ; WORD COUNT
3961          IBUFF          ; BUFFER ADDRESS
3962          .BYTE 0        2$:          ; SECTOR ADDRESS
3963          .BYTE 2        ; TRACK ADDRESS
3964          632           ; CYLINDER ADDRESS
3965
3966          TLOADRK          ;LOAD "L" REGS INTO RK
3967          TWAT112         ;WAIT FOR INTERRUPT
3968          ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
3969          TCHKOP          ;CHECK FOR ANY ERRORS
3970          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
3971          SCOPI          ;LOOP TO 1$ IF SW 9 SET
3972
3973          TSTB    $ERFLG          ;TEST FOR ERROR IN OPERATION
3974          BEQ     7$            ;NO-SKIP
3975          TST     R0             ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
3976          BNE    3$            ;NO-SKIP
3977          ADD     #2,2$          ;NEXT SECTOR ADDRESS
3978          CMPB   #10.,2$        ;PAST APPLICABLE SECTORS?
3979          BNE    6$            ;NO-SKIP
3980          MOV     #EM51,EMIN
3981          ERROR   1             ;"CANNOT READ BS FILES
3982          BIC    DRVBIT,$DEVMM   ;CLEAR DRIVE FROM DRIVE MAP
3983          JMP     NEWDRV         ;ABORT TEST PASS.
3984
3985          3$:          CMP     #1,R0          ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
3986          BNE    4$            ;NO-SKIP
3987          ADD     #2,2$          ;NEXT SECTOR ADDRESS
3988          CMPB   #22.,2$        ;PAST APPLICABLE SECTORS?
3989          BNE    6$            ;NO-SKIP
3990          MOV     #EM51,EMIN
3991          ERROR   1             ;"CANNOT READ BS FILES"

```


M06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 77
T20 READ BAD SECTOR INFORMATION

SEQ 0077

3992	007560	000754			BR	25\$	
3993							
3994	007562	022700	000002		4\$: CMP	#2,R0	;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
3995	007566	001014			BNE	5\$;NO-SKIP
3996	007570	062737	000002	007432	ADD	#2,2\$;NEXT SECTOR ADDRESS
3997	007576	122737	000013	007432	CMPB	#11.,2\$;PAST APPLICABLE SECTORS?
3998	007604	001021			BNE	6\$;NO-SKIP
3999							
4000	007606	012737	054647	001360	MOV	#EM51,EM1N	
4001	007614	104001			ERROR	1	; "CANNOT READ BS FILES"
4002	007616	000735			BR	25\$	
4003							
4004	007620	062737	000002	007432	5\$: ADD	#2,2\$;NEXT SECTOR (BS SOFT 24 SECT MODE)
4005	007626	122737	000027	007432	CMPB	#23.,2\$;PAST APPLICABLE SECTORS?
4006	007634	001005			BNE	6\$;NO-SKIP
4007	007636	012737	054647	001360	MOV	#EM51,EM1N	
4008	007644	104001			ERROR	1	; "CANNOT READ BS FILES"
4009	007646	000721			BR	25\$	
4010							
4011	007650	105037	001103		6\$: CLRB	\$ERFLG	;CLEAR ERROR FLAG
4012	007654	000657			BR	1\$;DO NEXT READ
4013							
4014	007656	005737	060610		7\$: TST	IBUFF+4	;CHECK FOR ALIGNMENT PACK
4015	007662	001405			BEQ	8\$;NO-SKIP
4016	007664	012737	054742	001360	MOV	#EM52,EM1N	
4017	007672	104001			ERROR	1	; "ALIGNMENT PACK. DRIVE ABORTING"
4018	007674	000706			BR	25\$	
4019							
4020	007676	012701	060614		8\$: MOV	#IBUFF+10,R1	;SET TO START OF BAD SECTOR DATA
4021							
4022	007702	022711	177777		9\$: CMP	#-1,(R1)	;TEST IF WORD ALL ONES (END OF DATA)
4023	007706	001417			BEQ	11\$;YES-SKIP
4024	007710	012123			MOV	(R1)+,(R3)+	;STORE CYLINDER
4025	007712	012123			MOV	(R1)+,(R3)+	;TRACK AND SECTOR
4026	007714	005205			INC	R5	;BUMP ERROR COUNTER
4027	007716	022705	000025		CMP	#21.,R5	;DOES IT TOTAL 20 FOR THIS FORMAT?
4028	007722	001367			BNE	9\$;NO-TEST AND MORE NEXT ADDRESS
4029	007724	012737	055020	001360	MOV	#EM53,EM1N	
4030	007732	104001			ERROR	1	;TO MANY BAD SECTORS
4031	007734	043737	001630	001354	10\$: BIC	DRVBIT,\$DEVN	;CLEAR DRIVE FROM TESTING
4032	007742	000137	027236		JMP	NEWDRV	;ABORT PASS
4033							
4034	007746	005200			11\$: INC	R0	;BUMP TO NEXT
4035	007750	022700	000001		CMP	#1,R0	;NOW TESTING BS SOFT 26 SECTOR MODE?
4036	007754	001011			BNE	12\$;NO-SKIP
4037	007756	012723	177777		MOV	#-1,(R3)+	;INSERT END OF FIELD FLAG
4038	007762	010337	001644		MOV	R3,BSS26P	;SET POINTER TO BAD SECTOR SOFTWARE FIELD
4039	007766	112737	000012	007432	MOVB	#12,2\$;SET TO FIRST SECTOR THIS MODE
4040	007774	000137	007414		JMP	1\$;GO READ IT.
4041	010000	022700	000002		12\$: CMP	#2,R0	;NOW TESTING BS FACT 24 SECTOR MODE?
4042	010004	001014			BNE	13\$;NO-SKIP
4043	010006	012723	177777		MOV	#-1,(R3)+	;INSERT END OF FIELD FLAG
4044	010012	112737	000001	007432	MOVB	#1,2\$;SET TO FIRST SECTOR THIS MODE
4045	010020	010537	001646		MOV	R5,BS26CT	;STORE TOTAL BS COUNT 26 SECTOR MODE
4046	010024	005005			CLR	R5	;CLEAR COUNTER FOR COUNTING 24 SECT BS
4047	010026	013703	001636		MOV	BSF24P,R3	;SET POINTER FOR STORING BS

N06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 78
T20 READ BAD SECTOR INFORMATION

SEQ 0078

```

4048 010032 000137 007414          JMP      1$          ;GO READ
4049
4050 010036 022700 000003          13$:   CMP      #3,R0          ;NOW TESTING BS SOFT 24 SECTOR MODE
4051 010042 001011                   BNE     14$          ;NO-SKIP
4052 010044 012723 177777          MOV     #-1,(R3)+    ;INSERT END OF FIELD FLAG
4053 010050 010337 001642          MOV     R3,BSS24P    ;STORE POINTER TO BSS 24 SECTOR MODE
4054 010054 112737 000013 007432  MOVB    #13,2$      ;GET START OF FIELDS BSS 24 SECT MODE
4055 010062 000137 007414          JMP     1$          ;GO READ IT
4056
4057 010066 012723 177777          14$:   MOV     #-1,(R3)+    ;INSERT END OF FIELD FLAG
4058 010072 010537 001650          MOV     R5,BS24CT    ;STORE COUNT BSS 24 SECTOR MODE
4059
4060
4061 010076 012700 057734          MOV     #BS26,R0     ;GET START OF BAD SECTOR BUFFER
4062 010102 012703 060144          MOV     #BS24+52.,R3 ;GET END OF BUFFER
4063
4064 010106 022720 000312          27$:   CMP     #312,(R0)+   ;TEST IF ANY SECTORS BAD IN CYL 312
4065 010112 001403                   BEQ     26$          ;YES - GET OUT
4066
4067 010114 020003                   CMP     R0,R3        ;CHECK IF ALL OF BUFFER TESTED
4068 010116 001373                   BNE     27$          ;NO - LOOP
4069 010120 000406                   BR      28$          ;EXIT
4070
4071 010122 012737 055701 001360  26$:   MOV     #DRVABT,EMIN ;"BAD SECTOR IN AREA FOR TEST"
4072 010130 104001                   ERROR   1
4073 010132 000137 007512          JMP     25$
4074
4075 010136          28$:
4076
4077          ;*****
4078          ;*TEST 21          FORMAT IN 26 SECTOR FORMAT
4079          ;*
4080          ;*          FORMAT CYLINDER 312, TRACK 0 AND TRACK 1 FOR 26 SECTOR
4081          ;*          FORMAT. VERIFY FORMAT AND THAT DATA LATE DID NOT
4082          ;*          OCCUR WITH WRITE HEADER OR IN READING DATA
4083          ;*          BUFFER AFTER READ HEADER.
4084          ;*
4085          ;*****
4086 010136 000004          †ST21: SCOPE
4087 010140 012737 000062 001262  MOV     #50.,$TIMES ;:DO 50. ITERATIONS
4088 010146 012737 000312 001664  MOV     #312,REFMT  ;:SET REFORMAT SWITCH
4089 010154 105037 010277          CLR    10$          ;:CLEAR SECTOR POINTER
4090 010160 105037 010203          CLR    11$          ;:CLEAR TRACK POINTER
4091 010164 104416          TSSINIT.           ;:CLEAR SUBSYSTEM
4092 010166 104003          ERROR    3          ;:BAD INIT ERROR
4093
4094 010170 004437 034574          9$:   JSR     R4,LRLOAD    ;LOAD "L" REGS
4095 010174 000127          WRHEAD           ;WRITE HEADER
4096 010176 177676          -102            ;WORD CNT FOR 26 SECTOR MODE
4097 010200 062604          OBUFF           ;BUFFER
4098 010202 000          .BYTE    0          ;SECTOR 0
4099 010203 000          .BYTE    0          ;TRACK 0
4100 010204 000312          312           ;CYL 0
4101
4102 010206 004437 040776          JSR     R4,GENCOM   ;GENERATE DATA
4103 010212 000600          000600        ;BUILD HEADERS-NO BAD SECTORS

```


4160 010426
 4161 010426 104416
 4162 010430 104003
 4163 010432 004437 035114
 4164 010436 104421
 4165 010440 104004
 4166 010442 104013
 4167 010444 104002

117\$:

```
TSSINIT          ;CLEAR SUBSYSTEM
ERROR            3 ;BAD INIT ERROR
JSR              R4,RDSTHD ;GO READ & SEQUENCE HEADERS
TCHKOP          ;CONTROLLER ERROR RETURN
ERROR           4 ; OR 5,6,7 ;REPORT ALL ERRORS
ERROR           13 ;"DATA LATE SET RESULT OF DATA BUFFER READ"
ERROR           2  ;"OPERATION TO SLOW" MESSAGE
                ;OR "HEADER D NOT FOUND" MESSAGE
```

4168
 4169
 4170 010446 004437 040776
 4171 010452 100200
 4172 010454 000414
 4173 010456 104015

7\$:

```
JSR              R4,GENCOM
100200          ;COMPARE IBUF & OBUF (HEADERS)
BR              6$ ;GOOD RETURN-NO MISCOMPARES
ERROR          15 ;REPORT 1ST MISCOMPARES
```

4174
 4175 010460 013700 001634
 4176 010464 005300
 4177 010466 001407
 4178 010470 004437 040776

12\$:

```
MOV             ERR_LMT,R0 ;GET ERROR LIMIT
DEC             R0         ;DECREMENT IT
BEQ            6$         ;EXIT IF ZERO
JSR            R4,GENCOM
```

4179
 4180 010474 040000
 4181 010476 000403
 4182 010500 104016
 4183 010502 000770
 4184 010504 104415

6\$:

```
040000          ;RESUME COMPARE
BR              6$         ;GOOD RETURN-NO MORE ERRORS
ERROR          16         ;REPORT NEXT ERROR LINE
BR              12$        ;LOOP
SCOPI          ;LOCAL ERROR LOOP TO 117$
```

4185
 4186 010506 105737 001607
 4187 010512 001010

8\$:

```
TSTB           L,DT      ;WAS TRACK 1 JUST DONE?
BNE            9$         ;YES-SKIP
```

4188
 4189 010514 112737 000001 010203
 4190 010522 112737 000001 010277
 4191 010530 000137 010170

```
MOVB           #1,11$    ;CHANGE PARAM TO LOAD "L" WITH
MOVB           #1,10$    ;TRACK 2
JMP            9$         ;JUMP TO DO ENTIRE TEST ON TRK 1
```

4192
4193 010534

8\$:

.SBTTL **HEADER RECOGNITION TESTS

4194
 4195
 4196
 4197
 4198
 4199
 4200
 4201
 4202
 4203
 4204
 4205
 4206
 4207
 4208
 4209
 4210
 4211
 4212
 4213 010534 000004
 4214 010536 012737 000062 001262
 4215 010544 012737 000312 001664

```
*****
;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.
*****
†T22:  SCOPE
      MOV     #50,$TIMES ;DO 50. ITERATIONS
      MOV     #312,REFMT ;SET REFORMAT SWITCH
```


E07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 82
T22 BAD SECTOR ERROR

SEQ 0092

4272 010734 105237 010666

INCB 5\$;BUMP TO NEXT SECTOR

4273 010740 000743

BR 4\$;LOOP

4274 010742 104421

6\$: TCHKOP ;CHECK FOR GOOD OPERATION

4276 010744 104004

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

4277 010746

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.

4289 010746 000004

TST23:

SCOPE ;DO 50. ITERATIONS
MOV #50., \$TIMES ;SET REFORMAT SWITCH
MOV #312, REFM T ;CLEAR SUBSYSTEM
TSSINIT ;BAD INIT ERROR
ERROR 3

4290 010750 012737 000062 001262

4291 010756 012737 000312 001664

4292 010764 104416

4293 010766 104003

4294 010770 004437 034574

JSR R4, LRLoad ;LOAD "L" REGS

4296 010774 000127

WRHEAD ;WRITE HEADER

4297 010776 177676

-102 ;WORD COUNT

4298 011000 062604

OBUFF ;BUFF ADD

4299 011002 000

.BYTE 0 ;SECT

4300 011003 000

.BYTE 0 ;TRACK

4301 011004 000312

312 ;CYL

4302 011006 004437 040776

JSR R4, GENCOM

4304 011012 000600

600 ;BUILD HEADERS NO BSE

4305 011014 012700 062610

MOV #OBUFF+4, R0 ;GET ADDRESS OF VRC HDR0

4307 011020 012703 000001

MOV #BIT0, R3 ;SET FOR BIT CHANGE SELECT

4308 011024 030310

1\$: BIT R3, (R0) ;CHECK A VRC BIT

4309 011026 001402

BEQ 2\$;SKIP IF ZERO

4310 011030 040310

BIC R3, (R0) ;ELSE CLEAR IT

4311 011032 000401

BR 3\$;SKIP

4312 011034 050310

2\$: BIS R3, (R0) ;IF ZERO SET IT

4313 011036 062700 000006

3\$: ADD #6, R0 ;BUMP TO NEXT VRC WORD

4314 011042 006303

ASL R3 ;SHIFT THE SELECT

4316 011044 001367

BNE 1\$;IF BIT NOT SHIFTED OUT-LOOP

4317 011046 104417

TLOADRK ;LOAD RK REGS

4319 011050 104431

TWAT112 ;WAIT FOR INTERRUPT

4320 011052 104002

ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

4321 011054 104421

TCHKOP ;CHECK OPERATION COMPLETE

4322 011056 104004

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

4323 011060 012737 011066 001110

4\$: MOV #4\$, \$LPERR ;SET LOCAL LOOP

4325 011066 104416

TSSINIT ;CLEAR SUBSYSTEM

4326 011070 104003

ERROR 3 ;BAD INIT ERROR

4327


```

4328 011072 004437 034574      JSR      R4,LRLOAD      ;LOAD "L" REGS
4329 011076 000123              WRDATA                ;WRITE DATA
4330 011100 177400              -400                  ;WORD COUNT
4331 011102 062604              OBUFF                 ;BUFFER ADD
4332 011104          000          5$: .BYTE 0              ;SECT
4333 011105          000          .BYTE 0              ;TRACK
4334 011106 000312              312                   ;CYL
4335
4336 011110 104417              TLOADRK               ;LOAD RK REG
4337 011112 104424              TWAT32                ;WAIT FOR INTERRUPT
4338 011114 104002              ERROR 2               ;TO SLOW/NOT COMPLETE ERROR
4339
4340 011116 022737 000020 011104  CMP      #16.,5$      ;WAS THIS WRITE SECTOR 16?
4341 011124 001415              BEQ      6$           ;YES-SKIP
4342
4343 011126 104422              TCHKWE                ;CHECK OPERATION WITH ERROR
4344 011130 000000              0
4345 011132 000040              40                    ;HVRC EM EXPECTED
4346 011134 000000              0
4347 011136 104004              ERROR 4 ; OR 5,6,7   ;REPORT ANY DISCREPENCIES
4348
4349 011140 104415              SCOPI                 ;LOCAL LOOP TO 4$
4350
4351 011142 105237 011104      INCB      5$          ;BUMP SECTOR IN "L" REG
4352 011146 022737 000016 011104  CMP      #16,5$      ;IF SECTOR IS 16 OR LESS
4353 011154 003744              BLE      4$          ;LOOP
4354 011156 000402              BR       7$          ;ELSE EXIT
4355 011160 104421          6$: TCHKOP            ;CHECK LAST OPERATION NO ERRORS
4356 011162 104004              ERROR 4 ; OR 5,6,7   ;REPORT ALL ERRORS
4357
4358 011164          7$:
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368 011164 000004          ;*****
4369 011166 012737 000062 001262  ;*TEST 24      BAD SECTOR ERROR AND HVRC ERROR
4370 011174 012737 000312 001664  ;*
4371 011202 104416          ;*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
4372 011204 104003          ;*      BOTH A BAD SECTOR ERROR AND HEADER VRC.  ISSUE A WRITE DATA
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382 011224 004437 040776      ;*
4383 011230 000600          ;*      TO CYLINDER 0, TRACK 0, SECTOR 0.  MAKE SURE ONLY HEADER VRC
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```



```

4384
4385 011232 042737 100000 062606      BIC      #BIT15,0BUFF+2 ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
4386
4387 011240 104417                      TLOADRK                      ;LOAD RK REGS
4388 011242 104431                      TWAT112                      ;WAIT FOR INTERRUPT
4389 011244 104002                      ERROR 2                      ;TO SLOW/NOT COMPLETE ERROR
4390
4391 011246 104421                      TCHKOP                       ;CHECK FOR ANY ERRORS
4392 011250 104004                      ERROR 4 ; OR 5,6,7          ;REPORT ALL ERRORS
4393
4394 011252 004437 034574              JSR      R4,LRLOAD           ;LOAD "L" REGS
4395 011256 000123                      WRDATA                       ;WRITE DATA
4396 011260 177400                      -400                         ;WORD COUNT
4397 011262 062604                      OBUFF                        ;BUFF ADD
4398 011264 000                      .BYTE 0                      ;SECTOR
4399 011265 000                      .BYTE 0                      ;TRACK
4400 011266 000312                      312                          ;CYLINDER
4401
4402 011270 104417                      TLOADRK                      ;LOAD RK REGS
4403 011272 104424                      TWAT32                      ;WAIT FOR INTERRUPT
4404 011274 104002                      ERROR 2                      ;TO SLOW/NOT COMPLETE ERROR
4405
4406 011276 104422                      TCHKWE                       ;CHECK OPERATION WITH EXPECTED ERR
4407 011300 000000                      0                             ;
4408 011302 000040                      40                          ;HVRC ERR EXPECTED
4409 011304 000000                      0                             ;
4410 011306 104004                      ERROR 4 ; OR 5,6,7          ;REPORT ALL DISCREPENCIES

```

```

*****
*TEST 25      OPERATION INCOMPLETE
*
*      FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE
*      WRONG FORMAT.  ISSUE A WRITE DATA OF 400 TO CYLINER 0,
*      TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
*
*****

```

```

4420 011310 000004                      ST25: SCOPE
4421 011312 012737 000062 001262      MOV      #50, $TIMES        ;DO 50. ITERATIONS
4422 011320 012737 000312 001664      MOV      #312,REFMT        ;SET REFORMAT SWITCH
4423 011326 104416                      TSSINIT                      ;CLEAR SUBSYSTEM
4424 011330 104003                      ERROR 3                      ;BAD INIT ERROR
4425
4426 011332 004437 034574              JSR      R4,LRLOAD           ;LOAD "L" REGS
4427 011336 000127                      WRHEAD                       ;WRITE HEADER
4428 011340 177676                      -102                         ;WORD COUNT FOR 26 SECT MODE
4429 011342 062604                      OBUFF                        ;BUFF ADD
4430 011344 000                      .BYTE 0                      ;SECTOR
4431 011345 000                      .BYTE 0                      ;TRACK
4432 011346 000312                      312                          ;CYLINDER
4433
4434 011350 004437 040776              JSR      R4,GENCOM          ;BUILD HEADERS-NO BSE ERRORS
4435 011354 000600                      600
4436
4437 011356 052737 001000 063004      BIS      #BIT9,0BUFF+200    ;CHANGE FORMAT IN SECTOR 25
4438 011364 052737 001000 063006      BIS      #BIT9,0BUFF+202    ;CORRECT THE VRC
4439

```



```

4440 011372 104417 TLOADRK ;LOAD RK REGS
4441 011374 104431 TWAT112 ;WAIT FOR INTERRUPT
4442 011376 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
4443
4444 011400 104421 TCHKOP ;CHECK FOR ANY ERRORS
4445 011402 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4446
4447 011404 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
4448 011410 000123 WRDATA ;WRITE DATA
4449 011412 177400 -400 ;400 WORDS
4450 011414 062604 OBUFF ;BUFF ADD
4451 011416 025 .BYTE 25 ;SECTOR 25
4452 011417 000 .BYTE 0 ;TRACK 0
4453 011420 000312 312 ;CYL 312
4454
4455 011422 104417 TLOADRK ;LOAD RK REGS
4456 011424 104425 TWAT48 ;WAIT FOR INTERRUPT
4457 011426 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
4458
4459 011430 104422 TCHKWE ;CHECK OPERATION EXPECTED ERROR
4460 011432 000000 0
4461 011434 000020 20 ;OPI EXPECTED
4462 011436 000000 0
4463 011440 104004 ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES

```

```

*****
*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.
*
*****

```

```

4473
4474 011442 000004 TST26: SCOPE
4475 011444 012737 000062 001262 MOV #50, $TIMES ;DO 50. ITERATIONS
4476 011452 012737 000312 001664 MOV #312,REFMT ;SET REFORMAT SWITCH
4477 011460 104416 TSSINIT ;CLEAR SUBSYSTEM
4478 011462 104003 ERROR 3 ;BAD INIT ERROR
4479
4480 011464 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
4481 011470 000127 WRHEAD ;WRITE HEADER
4482 011472 177676 -102 ;WORD COUNT FOR 26 SECT MODE
4483 011474 062604 OBUFF ;BUS ADDRESS
4484 011476 000 .BYTE 0 ;SECTOR
4485 011477 000 .BYTE 0 ;TRACK
4486 011500 000312 312 ;CYLINDER
4487
4488 011502 004437 040776 JSR R4,GENCOM
4489 011506 000600 600 ;BUILD HEADER- NO BSE ERRORS
4490
4491 011510 012700 062740 MOV #OBUFF+134,RO ;GET ADDRESS 2ND WORD HDR 17(8)
4492 011514 052720 001000 BIS #BIT9,(RO)+ ;SET FORMAT 24 SECT PER TRACK
4493 011520 052720 001000 BIS #BIT9,(RO)+ ;SET VRC BIT
4494 011524 062700 000004 ADD #4,RO ;BUMP TO HVRC WORD HDR 20(8)
4495 011530 032710 000001 BIT #BIT0,(RO) ;TEST BIT 0

```



```

4496 011534 001403          BEQ      1$          ;RESET-SKIP
4497 011536 042710 000001    BIC      #BIT0,(R0) ;CLEAR BIT
4498 011542 000402          BR       2$
4499 011544 052710 000001    1$:     BIS      #BIT0,(R0) ;SET BIT
4500                                ;FORCE OPI AND HVRC ERROR
4501 011550 104417          2$:     TLOADRK ;LOAD RK REGS
4502 011552 104431          TWAT112 ;WAIT FOR INTERRUPT
4503 011554 104002          ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
4504
4505 011556 104421          TCHKOP   ;CHECK FOR ANY ERRORS
4506 011560 104004          ERROR 4 ; OR 5,6,7 ;YES-REPORT ALL ERRORS
4507
4508 011562 004437 034574    JSR      R4,LRLOAD ;LOAD "L" REGS
4509 011566 000123          WRDATA   ;WRITE DATA
4510 011570 177400          -400    ;400 WORDS
4511 011572 062604          OBUFF   ;BUFF ADDRESS
4512 011574 017           .BYTE 17 ;SECT 17
4513 011575 000           .BYTE 0  ;TRACK 0
4514 011576 000312          312    ;CYLINDER 312
4515
4516 011600 104417          TLOADRK ;LOAD RK REGS
4517 011602 104425          TWAT48  ;WAIT FOR INTERRUPT
4518 011604 104002          ERROR 2    ;TO SLOW/NOT COMPLETE
4519
4520 011606 104422          TCHKWE   ;CHECK WITH EXPECTED ERROR
4521 011610 000000          0
4522 011612 000060          60    ;HVRC ERR & OPI EXPECTED
4523 011614 000000          0
4524 011616 104004          ERROR 4 ;OR 5,6,7
4525
4526 *****
4527 *TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
4528 *
4529 * FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
4530 * ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
4531 * AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
4532 * END OF THE OPERATION
4533 *****
4534 011620 000004          TST27:  SCOPE
4535 011622 012737 000100 001262  MOV      #100,$TIMES ;:DO 100 ITERATIONS
4536 011630 012737 000312 001664  MOV      #312,REFMT  ;SET REFORMAT SWITCH
4537
4538 011636 104416          TSSINIT ;CLEAR SUBSYSTEM
4539 011640 104003          ERROR 3 ;BAD INIT ERROR
4540
4541 011642 004437 034574    JSR      R4,LRLOAD ;LOAD "L" REGISTERS
4542 011646 000127          WRHEAD  ;WRITE HEADER
4543 011650 177676          -102   ;WORD COUNT FOR 26 SECTOR MODE
4544 011652 062604          OBUFF  ;BUFF ADD
4545 011654 000           .BYTE 0  ;SECTOR
4546 011655 000           .BYTE 0  ;TRACK
4547 011656 000312          312   ;CYLINDER
4548
4549 011660 004437 040776    JSR      R4,GENCOM
4550 011664 000600          600   ;BUILD HEADERS-NO BSE ERRORS
4551

```



```

4552 011666 012700 062750      MOV      #OBUF+144,R0      ;ADDRESS OF HEAD 20 HVRC WORD
4553 011672 012701 000002      MOV      #BIT1,R1         ;BIT 1 CONSTANT
4554 011676 030110              BIT      R1,(R0)          ;TEST BIT 1 SET
4555 011700 001402              BEQ      1$               ;RESET-SKIP
4556 011702 040110              BIC      R1,(R0)          ;ELSE CLEAR BIT 1
4557 011704 000401              BR       2$               ;SKIP
4558 011706 050110      1$:     BIS      R1,(R0)      ;SET BIT 1
4559
4560 011710 104417      2$:     TLOADRK          ;LOAD RK REGS
4561 011712 104431          TWAT112          ;WAIT FOR INTERRUPT
4562 011714 104002          ERROR 2          ;TO SLOW/NOT COMPLETE
4563
4564 011716 104421          TCHKOP          ;CHECK FOR ANY ERROR
4565 011720 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4566
4567 011722 004437 034574      JSR      R4,LRLOAD      ;LOAD "L" REGISTER
4568 011726 000123          WRDATA          ;WRITE DATA
4569 011730 177400          -400           ;WORD COUNT
4570 011732 062604          OBUF           ;BUFF ADD
4571 011734 021             .BYTE 21        ;SECTOR
4572 011735 000             .BYTE 0         ;TRACK
4573 011736 000312          312           ;CYLINDER
4574 011740 104417          TLOADRK          ;LOAD RK REGS
4575 011742 104424          TWAT32          ;WAIT FOR INTERRUPT
4576 011744 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
4577
4578 011746 104421          TCHKOP          ;CHECK FOR ANY ERROR
4579 011750 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS.
4580
4581
4582      .SBTTL  **DATA TRANSFER TESTS
4583
4584      ;*****
4585      ;*TEST 30      WRITE AND READ ONE SECTOR
4586      ;*
4587      ;*      FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
4588      ;*      TO AGREED WITH BAD SECTOR INFORMATION.  ISSUE A WRITE DATA
4589      ;*      OF ONE SECTOR ON CYLINDER 312, TRACK 0.  READ IT BACK TO
4590      ;*      MAKE SURE IT AGREES WITH WHAT IS WRITTEN.
4591      ;*
4592      ;*****
4593      TST30:  SCOPE
4594      MOV      #50, $TIMES      ;DO 50. ITERATIONS
4595      MOV      #312,REFMT      ;SET REFORMAT SWITCH
4596      TSSINIT          ;CLEAR SUBSYSTEM
4597      ERROR 3          ;BAD INIT ERROR
4598
4599      MOV      #312,7$         ;PRESET CYL POINTER
4600      CLRB  2$          ;CLEAR TRACK POINTER
4601
4602      1$:     JSR      R4,LRLOAD      ;LOAD "L" REG
4603      WRHEAD          ;WRITE HEADER
4604      -102           ;WORD COUNT FOR 26 SECTOR MODE
4605      OBUF           ;BUFF ADDRESS
4606      .BYTE 0         ;SECTOR
4607      2$:     .BYTE 0         ;TRACK

```


K07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.F11 01-OCT-76 13:08MACY11 27(1006) 05-OCT-76 09:17 PAGE 88
T30 WRITE AND READ ONE SECTOR

SEG 0088

4608	012022	000312		7\$:	312		;CYLINDER
4609							
4610	012024	004437	040776		JSR	R4,GENCOM	
4611	012030	001200			1200		;BUILD HDRS-INCLUDE BAD SECTORS
4612							
4613	012032	104417			TLOADRK		;LOAD RK REGS
4614	012034	104431			TWAT112		;WAIT FOR INTERRUPT
4615	012036	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4616							
4617	012040	104421			TCHKOP		;CHECK FOR ANY ERRORS
4618	012042	104004			ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS
4619							
4620	012044	022737	000313	012022	CMP	#313,7\$;TEST IF DONE 313 TK 0
4621	012052	001414			BEQ	3\$;YES - SKIP
4622	012054	123727	012021	000002	CMPB	2\$, #2	;DID WE JUST FORMAT TRACK 2
4623	012062	001403			BEQ	8\$;YES-SKIP
4624	012064	105237	012021		INCB	2\$;BUMP TO NEXT TRACK
4625	012070	000746			BR	1\$;GO FORMAT NEXT TRACK
4626							
4627	012072	105037	012021	8\$:	CLRB	2\$;CLEAR TRACK POINTER
4628	012076	005237	012022		INC	7\$;BUMP CYL TO 313
4629	012102	000741			BR	1\$;GO FORMAT 313 TK 0
4630							
4631	012104	004437	034574	3\$:	JSR	R4,LRLOAD	;LOAD "L" REGS
4632	012110	000123			WRDATA		;WRITE DATA
4633	012112	177400			-400		;ONE SECTOR WORD COUNT
4634	012114	062604			OBUFF		;BUFF ADDRESS
4635	012116	012			.BYTE	12	;SECTOR 12
4636	012117	000			.BYTE	0	;TRACK 0
4637	012120	000312			312		;CYLINDER 312
4638							
4639	012122	004437	040776		JSR	R4,GENCOM	
4640	012126	000001			1		;BUILD DATA PATTERN 1
4641	012130	000400			400		;400 WORDS LONG
4642	012132	012737	012140	001110	MOV	#4\$, \$LPERR	;SET FOR LOCAL LOOP
4643	012140	104417		4\$:	TLOADRK		;LOAD RK REGS
4644	012142	104431			TWAT112		;WAIT FOR INTERRUPT
4645	012144	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4646							
4647	012146	104421			TCHKOP		;CHECK FOR ANY ERRORS
4648	012150	104004			ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS
4649							
4650	012152	004437	034574		JSR	R4,LRLOAD	;LOAD "L" REGS
4651	012156	000121			RDDATA		;READ DATA
4652	012160	177400			-400		;400 WORDS
4653	012162	060604			IBUFF		;BUFF ADD
4654	012164	012			.BYTE	12	;SECTOR 12
4655	012165	000			.BYTE	0	;TRACK 0
4656	012166	000312			312		;CYL 312
4657							
4658	012170	104417			TLOADRK		;LOAD RK
4659	012172	104424			TWAT32		;WAIT FOR INTERRUPT
4660	012174	104002			ERROR	2	;TO SLOW/NOT COMPLETE
4661							
4662	012176	104421			TCHKOP		;CHECK FOR ANY ERRORS
4663	012200	104004			ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS


```

4664
4665 012202 004437 040776      JSR      R4,GENCOM
4666 012206 100001              100001      ;GO COMPARE DATA TO PATTERN 1
4667 012210 000400              400         ;400 WORDS LONG
4668 012212 000413              BR         6$ ;GOOD RETURN-NO DATA ERRORS
4669 012214 104015              ERROR      15 ;ERROR RETURN
4670
4671 012216 013700 001634      MOV      ERRLMT,RO ;GET ERROR LIMIT
4672 012222 005300      5$:      DEC      RO      ;DEC LIMIT
4673 012224 001406              BEQ      6$      ;EXIT IF 0
4674 012226 004437 040776      JSR      R4,GENCOM
4675 012232 040000              040000      ;CONTINUE COMPARE
4676 012234 000402              BR         6$      ;EXIT IF NO MORE ERRORS
4677 012236 104016              ERROR      16      ;ELSE REPORT MISCOMPARE
4678 012240 000770              BR         5$      ;LOOP
4679 012242 005037 001664      6$:      CLR      REFORMAT ;CLEAR REFORMAT SWITCH
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690 012246 000004
4691 012250 012737 000062 001262  *TEST 31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719

```

 *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.

```

*ST31: SCOPE
MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
TSSINIT
ERROR      3              ;CLEAR SUBSYSTEM
;BAD INIT ERROR
JSR      R4,LRLOAD      ;LOAD "L" REGS
WRDATA
-400      ;WRDATA
-400      ;-400 WORDS
OBUFF      ;OBUFF IS BUFF ADDRESS
.BYTE      12            ;SECTOR 12
.BYTE      1             ;TRACK 1
312       ;CYLINDER 312
BIS      #BAI,L,CS2      ;SET INCREMENT INHIBIT
JSR      R4,GENCOM      ;BUILD PATTERN
16        ;PATTERN 16
400       ;400 WORDS
TLOADRK
TWAT96
ERROR      2              ;LOAD RK REGS
;WAIT FOR INTERRUPT
;TO SLOW/NOT COMPLETE ERROR
TCHKOP
ERROR      4 ;OR 5, 6, 7, 10 ;CHECK OPERATION FOR ANY ERRORS
;REPORT ALL ERRORS
JSR      R4,LRLOAD      ;LOAD "L" REGS
RDATA
-400      ;RDATA
-400      ;-400 WORDS
IBUFF     ;IBUFF IS BUFF ADDRESS

```


M07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 90
T31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT

SEQ 0090

```

4720 012342 012 .BYTE 12 ;SECTOR 12
4721 012343 001 .BYTE 1 ;TRACK 1
4722 012344 000312 312 ;CYLINDER 312
4723
4724 012346 012700 000377 MOV #377,R0 ;SET COUNT TO SET OBUFF TO BE
4725 012352 012701 062606 MOV #OBUFF+2,R1 ;ALL THE FIRST WORD OF PATTERN
4726 012356 012703 062604 MOV #OBUFF,R3
4727
4728 012362 011321 1$: MOV (R3),(R1)+ ;MOV THE WORD
4729 012364 005300 DEC R0
4730 012366 001375 BNE 1$ ;LOOP UNTIL ALL WORDS SET
4731
4732 012370 104417 TLOADRK ;LOAD RK REGS
4733 012372 104424 TWAT32 ;WAIT FOR INTERRUPT
4734 012374 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4735
4736 012376 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4737 012400 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4738
4739 012402 004437 040776 JSR R4,GENCOM ;COMPARE THE DATA
4740 012406 100000 100000
4741 012410 000400 400
4742 012412 000413 BR 2$
4743 012414 104015 ERROR 15
4744 012416 013700 001634 MOV ERRMT,R0 ;GET ERROR LIMIT
4745 012422 005300 64$: DEC R0 ;DECREMENT COUNT
4746 012424 001406 BEQ 65$ ;IF ZERO - EXIT
4747 012426 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
4748 012432 040000 40000
4749 012434 000402 BR 65$ ;NO MORE ERRORS - EXIT
4750 012436 104016 ERROR 16 ;REPORT NEXT ERROR
4751 012440 000770 BR 64$ ;LOOP
4752 012442 65$:
4753
4754 012442 2$:
4755 ;*****
4756 ;*TEST 32 WRITE DATA ADDRESS GREATER THAN 32K
4757 ;*
4758 ;* ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.
4759 ;* MAKE SURE CORRECT DATA IS ON DISK.
4760 ;*
4761 ;* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K
4762 ;* OF MEMORY IS PRESENT.
4763 ;*
4764 ;*****
4765 012442 000004 TST32: SCOPE
4766 012444 012737 000062 001262 MOV #50,$TIMES ;DO 50. ITERATIONS
4767 012452 123727 001326 000001 CMPB $MAM$1,#1 ;TEST IF >32K MEM
4768 012460 002016 BGE 2$ ;YES-SKIP
4769
4770 012462 032737 000002 001656 BIT #MEMSZB,OPTFLG ;TEST IF REPORT ALREADY MADE
4771 012470 001011 BNE 1$ ;YES -SKIP
4772
4773 012472 104401 050411 TYPE ,OPR011 ;"INSUFFICIENT MEMORY DATA TRANSFER WITH
4774 012476 104401 050554 TYPE ,OPR012 ;ADDRESS >32K
4775 012502 052737 000002 001656 BIS #MEMSZB,OPTFLG ;SET FLAG

```


Address	OpCode	Operand 1	Operand 2	Operand 3	Label	Type	OPROIS	Comments
4776	012510	104401	050570			TYPE	OPROIS	:BYPASSED"
4777	012514	000470				BR	4\$:EXIT
4778	012516	012737	012524	001110		MOV	#5\$, \$LPERR	:SET LOCAL LOOP ON ERROR ADDRESS
4779	012524							
4780	012524	104416				TSSINIT		:CLEAR SUBSYSTEM
4781	012526	104003				ERROR	3	:BAD INIT ERROR
4782								
4783	012530	004437	034574			JSR	R4, LRLOAD	:LOAD "L" REGS
4784	012534	000123				WRDATA		:WRITE DATA
4785	012536	177400				-400		:400 WORDS
4786	012540	177770				177770		:BUS ADDRESS IN 32K -10 BYTES
4787	012542	016				.BYTE	16	:SECTOR 16
4788	012543	000				.BYTE	0	:TRACK 0
4789	012544	000312				312		:CYLINDER 312
4790	012546	004437	040776			JSR	R4, GENCOM	:GENERATE DATA
4791	012552	010010				10010		:PATTERN 10, MEM. MANAGEMENT FOR DEST.
4792	012554	001777				1777		:RELOCATION ARGUMENT
4793	012556	000400				400		:400 WORDS
4794								
4795	012560	104417				TLOADRK		:LOAD RK REGS
4796	012562	104430				TWAT96		:WAIT FOR INTERRUPT
4797	012564	104002				ERROR	2	:TO SLOW/NOT COMPLETE ERROR
4798								
4799	012566	104421				TCHKOP		:CHECK OPERATION FOR ANY ERRORS
4800	012570	104004				ERROR	4 ;OR 5, 6, 7,	10 :REPORT ALL ERRORS
4801	012572	104415				SCOPI		:LOCAL LOOP ON ERROR TO 5\$
4802								
4803	012574	004437	040776			JSR	R4, GENCOM	:CLEAR Ibuff TO 1'S.
4804	012600	002007				2007		
4805	012602	001000				1000		
4806								
4807	012604	004437	034574			JSR	R4, LRLOAD	:LOAD "L" REGS
4808	012610	000121				RDDATA		:RDDATA
4809	012612	177400				-400		:400 WORDS
4810	012614	060604				IBUFF		:IBUFF IS BUFF ADDRESS
4811	012616	016				.BYTE	16	:SECTOR 16
4812	012617	000				.BYTE	0	:TRACK 0
4813	012620	000312				312		:CYLINDER 312
4814	012622	104417				TLOADRK		:LOAD RK REGS
4815	012624	104424				TWAT32		:WAIT FOR INTERRUPT
4816	012626	104002				ERROR	2	:TO SLOW/NOT COMPLETE ERROR
4817	012630	104421				TCHKOP		:CHECK OPERATION FOR ANY ERRORS
4818	012632	104004				ERROR	4 ;OR 5, 6, 7,	10 :REPORT ALL ERRORS
4819	012634	004437	040776			JSR	R4, GENCOM	:COMPARE
4820	012640	110000				110000		:MEMORY MANAGEMENT FOR DESTINATION
4821	012642	001777				1777		:RELOCATION ARGUMENT
4822	012644	000400				400		:400 WORDS
4823	012646	000413				BR	4\$:NO ERROR-SKIP
4824	012650	104015				ERROR	15	:REPORT FIRST MISCOMPARE
4825	012652	013700	001634			MOV	ERRLMT, R0	:GET ERROR LIMIT
4826	012656	005300			64\$:	DEC	R0	:DECREMENT COUNT
4827	012660	001406				BEQ	65\$:IF ZERO - EXIT
4828	012662	004437	040776			JSR	R4, GENCOM	:CONTINUE DATA COMPARE
4829	012666	050000				50000		
4830	012670	000402				BR	65\$:NO MORE ERRORS - EXIT
4831	012672	104016				ERROR	16	:REPORT NEXT ERROR


```

4832 012674 000770
4833 012676
4834
4835 012676
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846 012676 000004
4847 012700 012737 000062 001262
4848 012706 123727 001326 000001
4849 012714 002001
4850
4851 012716 000462
4852
4853 012720 012737 012726 001110
4854
4855 012726
4856 012726 104416
4857 012730 104003
4858 012732 004437 034574
4859 012736 000123
4860 012740 177400
4861 012742 062604
4862 012744 017
4863 012745 000
4864 012746 000312
4865 012750 004437 040776
4866 012754 000011
4867 012756 000400
4868
4869 012760 104417
4870 012762 104430
4871 012764 104002
4872
4873 012766 104421
4874 012770 104004 034574
4875 012772 004437
4876 012776 000121
4877 013000 177400
4878 013002 177770
4879 013004 017
4880 013005 000
4881 013006 000312
4882
4883 013010 104417
4884 013012 104424
4885 013014 104002
4886 013016 104421
4887 013020 104004

```

```

655: BR 645 ;LOOP
45:
*****
*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 #50,$TIMES ;DO 50. ITERATIONS
CMPB $MAMS1,#1 ;CHECK IF >32K MEMORY
BGE 25 ;YES-SKIP
15: BR 55 ;EXIT
25: MOV #35,$LPERR ;SET LOCAL ERROR LOOP
35:
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
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" REG
RDATA ;READ DATA
-400 ;400 WORDS
177770 ;ACROSS 32K BOUNDARY
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYL 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

```



```

4888 013022 004437 040776      JSR      R4,GENCOM      ;COMPARE DATA
4889 013026 120000              120000      ;MEMORY MANAGEMENT WITH SOURCE
4890 013030 001777              1777        ;RELOCATION ARGUMENT
4891 013032 000400              400         ;COMPARE 400 WORDS
4892 013034 000413              BR         5$          ;NO MISCOMPARE-EXIT
4893 013036 104015              ERROR      15         ;REPORT FIRST MISCOMPARE
4894 013040 013700 001634      MOV        ERRLMT,R0    ;GET ERROR LIMIT
4895 013044 005300 64$:      DEC        R0          ;DECREMENT COUNT
4896 013046 001406              BEQ        65$         ;IF ZERO - EXIT
4897 013050 004437 040776      JSR      R4,GENCOM      ;CONTINUE DATA COMPARE
4898 013054 060000              60000      ;
4899 013056 000402              BR         65$         ;NO MORE ERRORS - EXIT
4900 013060 104016              ERROR      16         ;REPORT NEXT ERROR
4901 013062 000770              BR         64$         ;LOOP
4902 013064
4903 013064
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914 013064 000004
4915 013066 012737 000062 001262
4916 013074 123727 001326 000002
4917 013102 002016
4918 013104 032737 000002 001656 6$:
4919 013112 001011              BIT        #MEMSZB,OPTFLG ;TEST IF REPORT FLAG SET
4920
4921 013114 104401 050411              BNE        1$          ;NO-SKIP
4922 013120 104401 050560              TYPE      ,OPR011      ;"INSUFFICIENT MEMORY-DATA XFER WITH
4923 013124 104401 050570              TYPE      ,OPR013      ;ADDRESS >64K
4924 013130 052737 000002 001656 1$:      TYPE      ,OPR015      ;"BYPASSED"
4925 013136 000467              BIS        #MEMSZB,OPTFLG ;SET FLAG
4926
4927 013140 012737 013146 001110 2$:      BR         5$          ;
4928
4929 013146 3$:
4930 013146 104416              TSSINIT      ;CLEAR SUBSYSTEM
4931 013150 104003              ERROR      3          ;BAD INIT ERROR
4932 013152 004437 040776      JSR      R4,GENCOM      ;GENERATE DATA, PATTERN 11
4933 013156 010011              10011      ;MEM MANAGEMENT ON DESTINATION
4934 013160 003777              3777       ;RELOCATION ARGUMENT
4935 013162 000400              400        ;400 WORDS
4936
4937 013164 004437 034574      JSR      R4,LRLOAD      ;LOAD "L" REGS
4938 013170 000523              WRDATA:BA16 ;WRITE DATA AND SET BA16
4939 013172 177400              -400       ;400 WORDS
4940 013174 177770              177770     ;ACROSS 64K BOUNDARY
4941 013176 020              .BYTE      20         ;SECTOR 20
4942 013177 000              .BYTE      0          ;TRACK 0
4943 013200 000312              312        ;CYLINDER 312

```

```

65$:
5$:
*****
*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.
*****

```

```

†ST34: SCOPE
MOV      #50,$TIMES      ;DO 50. ITERATIONS
CMPB     $MAMS1,#2       ;CHECK IF >64K MEMORY
BGE      2$              ;YES-SKIP
BIT      #MEMSZB,OPTFLG  ;TEST IF REPORT FLAG SET
BNE      1$              ;NO-SKIP

"INSUFFICIENT MEMORY-DATA XFER WITH
ADDRESS >64K
"BYPASSED"
SET FLAG

;SET LOCAL LOOP ON ERROR

;CLEAR SUBSYSTEM
;BAD INIT ERROR
;GENERATE DATA, PATTERN 11
;MEM MANAGEMENT ON DESTINATION
;RELOCATION ARGUMENT
;400 WORDS

;LOAD "L" REGS
;WRITE DATA AND SET BA16
;400 WORDS
;ACROSS 64K BOUNDARY
;SECTOR 20
;TRACK 0
;CYLINDER 312

```



```

4944
4945 013202 104417 TLOADRK ;LOAD RK REGS
4946 013204 104430 TWAT96 ;WAIT FOR INTERRUPT
4947 013206 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4948
4949 013210 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4950 013212 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4951 013214 004437 040776 JSR R4,GENCOM ;CLEAR Ibuff TO 1'S
4952 013220 002007
4953 013222 001000
4954
4955 013224 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
4956 013230 000121 RDDATA ;RDDATA
4957 013232 177400 -400 ;-400 WORDS
4958 013234 060604 Ibuff ;IBUFF IS BUFF ADDRESS
4959 013236 020 .BYTE 20 ;SECTOR 20
4960 013237 000 .BYTE 0 ;TRACK 0
4961 013240 000312 312 ;CYLINDER 312
4962 013242 104417 TLOADRK ;LOAD RK REGS
4963 013244 104424 TWAT32 ;WAIT FOR INTERRUPT
4964 013246 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4965
4966 013250 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
4967 013252 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4968 013254 004437 040776 JSR R4,GENCOM ;CHECK DATA
4969 013260 110000 ;MEMORY MANAGEMENT WITH DESTINATION
4970 013262 003777 3777 ;RELOCATION ARGUMENT
4971 013264 000400 400 ;400 WORDS
4972 013266 000413 BR 5$ ;NO MISCOMPARES-SKIP
4973 013270 104015 ERROR 15 ;REPORT FIRST ERROR
4974
4975 013272 013700 001634 MOV ERLMT,RO ;GET ERROR LIMIT
4976 013276 005300 64$: DEC RO ;DECREMENT COUNT
4977 013300 001406 BEQ 65$ ;IF ZERO - EXIT
4978 013302 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
4979 013306 050000 50000
4980 013310 000402 BR 65$ ;NO MORE ERRORS - EXIT
4981 013312 104016 ERROR 16 ;REPORT NEXT ERROR
4982 013314 000770 BR 64$ ;LOOP
4983 013316
4984 013316
4985
4986 *****
4987 ;*TEST 35 READ DATA ADDRESS GREATER THAN 64K
4988 ;*
4989 ;* ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 377770.
4990 ;* CHECK MEMORY FOR CORRECT TRANSFER.
4991 ;*
4992 ;* NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K
4993 ;* OF MEMORY IS PRESENT.
4994 ;*
4995 *****
4996 TST35: SCOPE
4997 MOV #50,$TIMES ;DO 50 ITERATIONS
4998 CMPB $MAMS1,#2 ;CHECK IF >64K MEMORY
4999 BGE 2$ ;YES-SKIP
5000 BR 5$ ;EXIT

```


E08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 95
T35 READ DATA ADDRESS GREATER THAN 64K

SEQ 0095

```

5000
5001 013340 012737 000032 001110 2$: MOV #32,$LPERR ;SET LOCAL LOOP ON ERROR
5002
5003 013346 3$:
5004 013346 104416 TSSINIT ;CLEAR SUBSYSTEM
5005 013350 104003 ERROR 3 ;BAD INIT ERROR
5006 013352 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
5007 013356 000123 WRDATA ;WRDATA
5008 013360 177400 -400 ;-400 WORDS
5009 013362 062604 OBUFF ;OBUFF IS BUFF ADDRESS
5010 013364 021 .BYTE 21 ;SECTOR 21
5011 013365 000 .BYTE 0 ;TRACK 0
5012 013366 000312 312 ;CYLINDER 312
5013 013370 004437 040776 JSR R4,GENCOM ;GENERATE DATA
5014 013374 000012 12 ;PATTERN 12
5015 013376 000400 400 ;400 WORDS
5016
5017 013400 104417 TLOADRK ;LOAD RK REGS
5018 013402 104430 TWAT96 ;WAIT FOR INTERRUPT
5019 013404 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5020
5021 013406 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5022 013410 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5023 013412 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
5024 013416 000521 RDDATA:BA16 ;READ DATA AND SET BA16
5025 013420 177400 -400 ;400 WORDS
5026 013422 177770 177770 ;ACROSS 64K BOUNDARY
5027 013424 021 .BYTE 21 ;FROM SECTOR 21
5028 013425 000 .BYTE 0 ;TRACK 0
5029 013426 000312 312 ;CYLINDER 312
5030
5031 013430 104417 TLOADRK ;LOAD RK REGS
5032 013432 104424 TWAT32 ;WAIT FOR INTERRUPT
5033 013434 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5034
5035 013436 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5036 013440 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5037 013442 004437 040776 JSR R4,GENCOM ;COMPARE DATA
5038 013446 120000 ;MEM MANAGEMENT WITH SOURCE
5039 013450 003777 3777 ;RELOCATION ARGUMENT
5040 013452 000400 400 ;400 WORDS
5041 013454 000413 BR 5$ ;NO MISCOMPARES-SKIP
5042 013456 104015 ERROR 15 ;REPORT FIRST ERROR
5043
5044 013460 013700 001634 MOV ERRLMT,R0 ;GET ERROR LIMIT
5045 013464 005300 64$: DEC R0 ;DECREMENT COUNT
5046 013466 001406 BEQ 65$ ;IF ZERO - EXIT
5047 013470 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5048 013474 060000 60000
5049 013476 000402 BR 65$ ;NO MORE ERRORS - EXIT
5050 013500 104016 ERROR 16 ;REPORT NEXT ERROR
5051 013502 000770 BR 64$ ;LOOP
5052
5053 65$:
5054 013504 5$:
5055 ;:*****

```


F08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 96
T36 WRITE DATA ADDRESS GREATER THAN 96K

SEQ 0096

```

5056 : *TEST 36 WRITE DATA ADDRESS GREATER THAN 96K
5057 : *
5058 : * ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.
5059 : * MAKE SURE CORRECT DATA IS ON DISK.
5060 : *
5061 : * NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5062 : * OF MEMORY IS PRESENT.
5063 : *
5064 : *****
5065 013504 000004 TST36: SCOPE
5066 013506 012737 000062 001262 MOV #50,$TIMES ;DO 50. ITERATIONS
5067 013514 123727 001326 000003 CMPB $MAMS1,#3 ;CHECK IF >96K MEMORY
5068 013522 002016 BGE 3$ ;YES-SKIP
5069 013524 032737 000002 001656 1$: BIT #MEMSZB,OPTFLG ;TEST IF REPORT FLAG SET
5070 013532 001011 BNE 2$ ;NO-SKIP
5071
5072 013534 104401 050411 TYPE ,OPR011 ;"INSUFFICIENT MEMORY-DATA TRANSFET WITH
5073 013540 104401 050564 TYPE ,OPR014 ;ADDRESS >96K BYPASSED"
5074 013544 104401 050570 TYPE ,OPR015
5075 013550 052737 000002 001656 BIS #MEMSZB,OPTFLG ;SET REPORT FLAG
5076 013556 000463 BR 6$
5077
5078 013560 012737 013566 001110 3$: MOV #4$,$LPERR ;SET LOCAL LOOP ON ERROR
5079
5080 013566 4$:
5081 013566 104416 TSSINIT ;CLEAR SUBSYSTEM
5082 013570 104003 ERROR 3 ;BAD INIT ERROR
5083 013572 004437 034574 JSR R4,LRLOAD ;LOAD "L" REG
5084 013576 001123 WRDATA:BA17 ;WRITE DATA AND BA17
5085 013600 177400 -400 ;400 WORDS FROM
5086 013602 177770 177770 ;ACROSS 96K BOUNDARY
5087 013604 022 .BYTE 22 ;TO SECTOR 22
5088 013605 000 .BYTE 0 ;TRACK 0
5089 013606 000312 312 ;CYL 312
5090 013610 004437 040776 JSR R4,GENCOM ;GENERATE DATA
5091 013614 010013 10013 ;PATTERN 13 MEM MAN WITH DEST.
5092 013616 005777 5777 ;RELOCATION ARGUMENT
5093 013620 000400 400 ;400 WORDS
5094
5095 013622 104417 TLOADRK ;LOAD RK REGS
5096 013624 104430 TWAT96 ;WAIT FOR INTERRUPT
5097 013626 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5098
5099 013630 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5100 013632 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5101
5102 013634 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
5103 013640 000121 RDDATA ;RDDATA
5104 013642 177400 -400 ;-400 WORDS
5105 013644 060604 Ibuff ;IBUFF IS BUFF ADDRESS
5106 013646 022 .BYTE 22 ;SECTOR 22
5107 013647 000 .BYTE 0 ;TRACK 0
5108 013650 000312 312 ;CYLINDER 312
5109
5110 013652 104417 TLOADRK ;LOAD RK REGS
5111 013654 104424 TWAT32 ;WAIT FOR INTERRUPT

```



```

S112 013656 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
S113
S114 013660 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
S115 013662 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
S116
S117 013664 004437 040776    JSR R4,GENCOM   ;COMPARE DATA
S118 013670 110000          110000         ;MEM MANAGEMENT WITH DESTINATION
S119 013672 005777          5777           ;RELOCATION ARGUMENT
S120 013674 000400          400            ;400 WORDS
S121 013676 000413          BR 6$          ;NO MISCOMPARES-BRANCH
S122 013700 104015          ERROR 15       ;REPORT 1ST ERROR
S123
S124 013702 013700 001634    MOV ERRMT,RO   ;GET ERROR LIMIT
S125 013706 005300          64$: DEC RO     ;DECREMENT COUNT
S126 013710 001406          BEQ 65$       ;IF ZERO - EXIT
S127 013712 004437 040776    JSR R4,GENCOM ;CONTINUE DATA COMPARE
S128 013716 050000          50000
S129 013720 000402          BR 65$       ;NO MORE ERRORS - EXIT
S130 013722 104016          ERROR 16      ;REPORT NEXT ERROR
S131 013724 000770          BR 64$       ;LOOP
S132
S133
S134 013726
S135
S136
S137
S138
S139
S140
S141
S142
S143
S144
S145 013726 000004          TST37: SCOPE
S146 013730 012737 000062 001262 MOV #50,$TIMES ;:DO 50. ITERATIONS
S147 013736 123727 001326 000003 CMPB $MAMS1,#3 ;CHECK IF >96K MEMORY
S148 013744 002001          BGE 3$        ;YES-SKIP
S149 013746 000462          2$: BR 6$
S150
S151 013750 012737 013756 001110 3$: MOV #4$,$LPERR ;SET LOCAL LOOP ON ERROR
S152
S153
S154 013756          4$: TSSINIT
S155 013756 104416          ERROR 3       ;CLEAR SUBSYSTEM
S156 013760 104003          ;BAD INIT ERROR
S157 013762 004437 034574    JSR R4,LRLOAD ;LOAD "L" REGS
S158 013766 000123          WRDATA        ;WRDATA
S159 013770 177400          -400         ;-400 WORDS
S160 013772 062604          OBUFF        ;OBUFF IS BUFF ADDRESS
S161 013774 005           .BYTE 5       ;SECTOR 5
S162 013775 000           .BYTE 0       ;TRACK 0
S163 013776 000312          312         ;CYLINDER 312
S164 014000 004437 040776    JSR R4,GENCOM ;GENERATE DATA
S165 014004 000014          14          ;PATTERN 14
S166 014006 000400          400         ;400 WORDS
S167

```


H08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 98
T37 READ DATA ADDRESS GREATER THAN 96K

SEQ 0098

```

5168 014010 104417 TLOADRK ;LOAD RK REGS
5169 014012 104430 TWAT96 ;WAIT FOR INTERRUPT
5170 014014 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5171
5172 014016 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5173 014020 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5174 014022 004437 034574 JSR R4,LLOAD ;LOAD "L" REGS
5175 014026 001121 RDATA:BA17 ;READ DATA WITH BA17 SET
5176 014030 177400 -400 ;400 WORDS
5177 014032 177770 177770 ;ACROSS 96K BOUNDARY
5178 014034 005 .BYTE 5 ;FROM SECTOR 5
5179 014035 000 .BYTE 0 ;TRACK 0
5180 014036 000312 312 ;CYL 312
5181
5182 014040 104417 TLOADRK ;LOAD RK REGS
5183 014042 104424 TWAT32 ;WAIT FOR INTERRUPT
5184 014044 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5185
5186 014046 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5187 014050 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5188 014052 004437 040776 JSR R4,GENCOM ;COMPARE DATA
5189 014056 120000 120000 ;MEM MANAGEMENT WITH SOURCE
5190 014060 005777 5777 ;RELOCATION ARGUMENT
5191 014062 000400 400 ;400 WORDS
5192 014064 000413 BR 6$ ;NO MISCOMPARES-SKIP
5193 014066 104015 ERROR 15 ;REPORT FIRST ERROR
5194
5195 014070 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
5196 014074 005300 64$: DEC RO ;DECREMENT COUNT
5197 014076 001406 BEQ 65$ ;IF ZERO - EXIT
5198 014100 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5199 014104 060000 60000
5200 014106 000402 BR 65$ ;NO MORE ERRORS - EXIT
5201 014110 104016 ERROR 16 ;REPORT NEXT ERROR
5202 014112 000770 BR 64$ ;LOOP
5203 014114
5204
5205 014114
5206
5207
5208
5209
5210
5211
5212
5213
5214 014114 000004 ST40: SCOPE
5215 014116 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
5216 014124 104416 TSSINIT ;CLEAR SUBSYSTEM
5217 014126 104003 ERROR 3 ;BAD INIT ERROR
5218
5219 014130 004437 034574 JSR R4,LLOAD ;LOAD "L" REG
5220 014134 000123 WRDATA ;WRITE DATA
5221 014136 177675 -103 ;WORD COUNT PARTIAL SECTOR
5222 014140 062604 OBUF ;BUFF ADDRESS
5223 014142 007 .BYTE 7 ;SECTOR 7

```

```

*****
*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.
*****

```



```

5224 014143 000 .BYTE 0 ;TRACK 0
5225 014144 000312 312 ;CYLINDER 312
5226
5227 014146 004437 040776 JSR R4,GENCOM ;GENERATE DATA
5228 014152 000003 3 ;PATTERN 3
5229 014154 000400 400 ;400 WORDS
5230
5231 014156 104417 TLOADRK ;LOAD RK REGS
5232 014160 104430 TWAT96 ;WAIT FOR INTERRUPT
5233 014162 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5234
5235 014164 104421 TCHKOP ;CHECK FOR ANY ERROR
5236 014166 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERROR
5237
5238 014170 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
5239
5240 014174 000121 RDDATA ;READ DATA
5241 014176 177400 -400 ;ONE FULL SECTOR
5242 014200 060604 IBUFF ;BUFF ADDRESS
5243 014202 007 .BYTE 7 ;SECTOR 7
5244 014203 000 .BYTE 0 ;TRACK 0
5245 014204 000312 312 ;CYLINDER 312
5246
5247 014206 004437 040776 JSR R4,GENCOM
5248 014212 002007 2007 ;CLEAR IBUFF TO ALL ONES
5249 014214 000400 400
5250
5251 014216 104417 TLOADRK ;LOAD RK REGS
5252 014220 104424 TWAT32 ;WAIT FOR INTERRUPT
5253 014222 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
5254
5255 014224 104421 TCHKOP ;CHECK FOR ANY ERRORS
5256 014226 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
5257
5258 014230 012701 063012 MOV #OBUFF+206,R1 ;CLEAR THE LAST 205 WORDS
5259 014234 012700 000275 MOV #275,R0 ;OF THE OUTPUT BUFFER TO ZERO
5260 014240 005021 1$: CLR (R1)+ ;TO VERIFY THE PARTIAL SECTOR
5261 014242 005300 DEC R0 ;WRITE 0 FILLED THE SECTOR
5262 014244 001375 BNE 1$
5263 014246 004437 040776 JSR R4,GENCOM
5264 014252 100000 100000 ;COMPARE OBUFF & IBUFF.
5265 014254 000400 400 ;ALL 400 WORDS
5266 014256 000413 BR 3$ ;NO ERRORS-EXIT
5267 014260 104015 ERROR 15 ;REPORT FIRST COMPARE ERROR
5268
5269 014262 013700 001634 2$: MOV ERRLMT,R0 ;GET ERROR LIMIT
5270 014266 005300 DEC R0 ;DECREMENT IT
5271 014270 001406 BEQ 3$ ;IF ZERO-EXIT
5272 014272 004437 040776 JSR R4,GENCOM
5273 014276 040000 40000 ;CONTINUE COMPARE
5274 014300 000402 BR 3$ ;NO MORE ERRORS-EXIT
5275 014302 104016 ERROR 16 ;REPORT NEXT COMPARE ERROR
5276 014304 000770 BR 2$ ;LOOP
5277
5278 014306 3$:
5279 ;:*****

```


JOB

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 100
T41 PARTIAL SECTOR READ DATA

SEQ 0100

```

5280 ;*TEST 41 PARTIAL SECTOR READ DATA
5281 ;*
5282 ;* WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A
5283 ;* KNOWN CONFIGURATION. ISSUE A READ DATA OF
5284 ;* 103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0.
5285 ;* MAKE SURE ONLY 103 WORDS GET TRANSFERRED
5286 ;* TO MEMORY.
5287 ;*
5288 ;*
5289 *****
5289 014306 000004 TST41: SCOPE
5290 014310 012737 000062 001262 MOV #50.,$TIMES ;:DO 50. ITERATIONS
5291 014316 104416 TSSINIT ;:CLEAR SUBSYSTEM
5292 014320 104003 ERROR 3 ;:BAD INIT ERROR
5293
5294 014322 004437 034574 JSR R4,LRLOAD ;:LOAD "L" REGS
5295 014326 000123 WRDATA ;:WRDATA
5296 014330 177400 -400 ;:-400 WORDS
5297 014332 062604 OBUFF ;:OBUFF IS BUFF ADDRESS
5298 014334 017 .BYTE 17 ;:SECTOR 17
5299 014335 000 .BYTE 0 ;:TRACK 0
5300 014336 000312 312 ;:CYLINDER 312
5301 014340 004437 040776 JSR R4,GENCOM ;:GENERATE DATA
5302 014344 000004 4 ;:PATTERN 4
5303 014346 000400 400 ;:400 WORDS
5304
5305 014350 104417 TLOADRK ;:LOAD RK REGS
5306 014352 104430 TWAT96 ;:WAIT FOR INTERRUPT
5307 014354 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
5308
5309 014356 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
5310 014360 104004 ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS
5311
5312 014362 004437 034574 JSR R4,LRLOAD ;:LOAD "L" REGS
5313 014366 000121 RDDATA ;:RDDATA
5314 014370 177675 -103 ;:-103 WORDS
5315 014372 060604 IBUFF ;:IBUFF IS BUFF ADDRESS
5316 014374 017 .BYTE 17 ;:SECTOR 17
5317 014375 000 .BYTE 0 ;:TRACK 0
5318 014376 000312 312 ;:CYLINDER 312
5319 014400 004437 040776 JSR R4,GENCOM
5320 014404 002007 2007 ;:CLEAR IBUFF
5321 014406 000400 400
5322
5323 014410 104417 TLOADRK ;:LOAD RK REGS
5324 014412 104424 TWAT32 ;:WAIT FOR INTERRUPT
5325 014414 104002 ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
5326 014416 104421 TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
5327 014420 104004 ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS
5328
5329 014422 012700 063012 MOV #OBUFF+206,R0 ;:AFTER THE LAST 205 WORDS OF
5330 014426 012701 000275 MOV #275,R1 ;:THE OUTPUT BUFFER TO ALL ONES.
5331 014432 012720 177777 1$: MOV #-1,(R0)+ ;:THESE SHOULD ALL BE ONES IN
5332 014436 005301 DEC R1 ;:IBUFF BECAUSE THE PARTIAL
5333 014440 001374 BNE 1$ ;:READ FILLED ONLY 103 WORDS.
5334 014442 004437 040776 JSR R4,GENCOM ;:GO COMPARE IBUFF & OBUFF
5335 014446 100000 100000

```


K08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 101
T41 PARTIAL SECTOR READ DATA

SEQ 0101

```

5336 014450 000400          400          ;ALL 400 WORDS
5337 014452 000413          BR          3$          ;NO ERRORS-EXIT
5338 014454 104015          ERROR       15          ;REPORT FIRST COMPARE ERROR
5339
5340 014456 013700 001634          MOV          ERRHLT,RO  ;GET ERROR LIMIT
5341 014462 005300          64$: DEC          RO          ;DECREMENT COUNT
5342 014464 001406          BEQ          65$          ;IF ZERO - EXIT
5343 014466 004437 040776          JSR          R4,GENCOM  ;CONTINUE DATA COMPARE
5344 014472 040000          40000
5345 014474 000402          BR          65$          ;NO MORE ERRORS - EXIT
5346 014476 104016          ERROR       16          ;REPORT NEXT ERROR
5347 014500 000770          BR          64$          ;LOOP
5348 014502          65$:
5349
5350 014502          3$:
5351          ;*****
5352          ;*TEST 42          WRITE DATA WITH NON-EXISTENT MEMORY
5353          ;*
5354          ;*          ISSUE A WRITE DATA OF 1 WORD USING ADDRESS 776000.
5355          ;*          MAKE SURE NON-EXISTENT MEMORY SETS.
5356          ;*
5357          ;*****
5358 014502 000004          TST42: SCOPE
5359 014504 012737 000062 001262          MOV          #50.,$TIMES ;DO 50. ITERATIONS
5360 014512 104416          TSSINIT
5361 014514 104003          ERROR       3          ;CLEAR SUBSYSTEM
5362          ;BAD INIT ERROR
5363 014516 004437 034574          JSR          R4,LRLOAD  ;LOAD "L" REG
5364 014522 001523          BA16!BA17!WRDATA      ;BA16 & 17 SET WITH WRITE DATA
5365 014524 177777          -1          ;WORD COUNT OF 1
5366 014526 176000          176000      ;BUFF ADDRESS=IO PAGE BASE
5367 014530          013        ;SECT 13
5368 014531          000        ;TRACK 0
5369 014532 000312          312        ;CYLINDER 312
5370
5371 014534 104417          TLOADRK
5372 014536 104430          TWAT96
5373 014540 104002          ERROR       2          ;LOAD RK REGS
5374 014542 104422          TCHKWE
5375 014544 000000          0          ;WAIT FOR INTERRUPT
5376 014546 000000          0          ;TO SLOW/NOT COMPLETE ERROR
5377 014550 000040          NEMERR
5378 014552 104004          ERROR       4 ;OR5,6,7 ;CHECK OPERATION WITH ERROR
5379 014554 012737 052762 001450          MOV          #EM11A,EM10N ;NON-EXISTENT MEMORY ERROR
5380 014562 012737 047276 057662          MOV          #OPER42,DF010A ;REPORT ANY DISCREPENCIES
5381 014570 113700 001541          MOV          T.CS1+1,RO  ;SET UP ERROR MESSAGE
5382 014574 042700 177774          BIC          #177774,RO  ;WITH SUPPORT MESSAGE
5383 014600 022700 000003          CMP          #3,RO      ;GET UPPER CS1
5384 014604 001406          BEQ          1$          ;CLEAR UNUSED BITS
5385 014606 010037 001204          MOV          RO,$REG11  ;TEST IF BOTH UPPER BUS BITS SET
5386 014612 012737 000003 001202          MOV          #3,$REG10  ;YES - SKIP
5387 014620 104010          ERROR       10         ;SET UP FOR ERROR REPORT
5388 014622 022737 176002 001544 1$: CMP          #176002,T.BA ;TEST IF BUSS ADDRESS LOW OKAY
5389 014630 001412          BEQ          2$          ;YES - SKIP
5390 014632 012737 052734 001450          MOV          #EM11,EM10N ;SET UP MESSGAE
5391 014640 012737 176002 001202          MOV          #176002,$REG10 ;STORE VALUE FOR REPORT

```


L08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 102
T42 WRITE DATA WITH NON-EXISTENT MEMORY

SEQ 0102

```

5392 014646 013737 001544 001204      MOV    T.BA,$REG11
5393 014654 104010                      ERROR  10
5394
5395 014656 005737 001542      2$:   TST    T.WC          ;TEST IF WORD COUNT CORRECT
5396 014662 001411                      BEQ    3$              ;YES - SKIP
5397 014664 012737 052707 001450      MOV    #EM10,EM10N    ;SET UP MESSAGE
5398 014672 005037 001202                      CLR    $REG10
5399 014676 013737 001542 001204      MOV    T.WC,$REG11
5400 014704 104010                      ERROR  10
5401
5402 014706      3$:
5403
5404
5405
5406
5407
5408
5409
5410
5411 014706 000004      *****
5412 014710 012737 000062 001262      *TEST 43      READ DATA WITH NON-EXISTENT MEMORY
5413 014716 104416                      *
5414 014720 104003                      *      ISSUE A READ DATA OF 1 WORD USING ADDRESS 776000.
5415
5416 014722 004437 034574                      *      MAKE SURE NON-EXISTENT MEMORY SETS.
5417 014726 001521                      *
5418 014730 177777                      *****
5419 014732 176000      TST43:  SCOPE
5420 014734 013          MOV    #50.,$TIMES    ;DO 50. ITERATIONS
5421 014735 000          TSSINIT              ;CLEAR SUBSYSTEM
5422 014736 000312      ERROR  3              ;BAD INIT ERROR
5423
5424 014740 104417      JSR    R4,LRLOAD     ;LOAD "L" REG
5425 014742 104430      BA16!BA17!RDATA     ;BA16 & 17 WITH READ DATA
5426 014744 104002      -1                  ;WORD COUNT OF 1
5427 014746 104422      176000              ;BUFF ADDRESS=IO PAGE BASE
5428 014750 000000      .BYTE 13            ;SECTOR 13
5429 014752 000000      .BYTE 0              ;TRACK 0
5430 014754 000040      312                  ;CYL 312
5431 014756 104004      TLOADRK              ;LOAD RK REGS
5432 014760 012737 052762 001450      TWAT96              ;WAIT FOR INTERRUPT
5433 014766 012737 047352 057662      ERROR  2              ;TO SLOW/NOT COMPLETE ERROR
5434 014774 113700 001541      TCHKWE              ;CHECK OPERATION WITH ERRORS
5435 015000 042700 177774      0
5436 015004 022700 000003      0
5437 015010 001406      NEMERR              ;NON-EXISTENT MEMORY ERROR
5438 015012 012737 000003 001202      ERROR  4: OR 5,6,7    ;REPORT ALL DISCREPANCIES
5439 015020 010037 001204      MOV    #EM11A,EM10N ;SET MESSAGE
5440 015024 104010      MOV    #OPER43,DF010A ;SET SUPPORT MESSAGE
5441
5442 015026 022737 176002 001544      1$:   MOV    7.CS1+1,R0    ;GET UPPER CS1
5443 015034 001412      MOV    #177774,R0    ;CLEAR UNWANTED BITS
5444 015036 012737 052734 001450      BIC    #3,R0          ;TEST BOTH BUS 16 & 17 SET
5445 015044 012737 176002 001202      CMP    #3,R0          ;YES - SKIP
5446 015052 013737 001544 001204      BEQ    1$              ;SET VALUES FOR REPORT
5447 015060 104010      MOV    #3,$REG10
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024
6025
6026
6027
6028
6029
6030
6031
6032
6033
6034
6035
6036
6037
6038
6039
6040
6041
6042
6043
6044
6045
6046
6047
6048
6049
6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192
6193
6194
6195
6196
6197
6198
6199
6200
6201
6202
6203
6204
6205
6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429
6430
6431
6432
6433
6434
6435
6436
6437
6438
6439
6440
6441
6442
6443
6444
6445
6446
6447
6448
6449
6450
6451
6452
6453
6454
6455
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633
6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656
6657
6658
6659
6660
6661
6662
6663
6664
6665
6666
6667
6668
6669
6670
6671
6672
6673
6674
6675
6676
6677
6678
6679
6680
6681
6682
6683
6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858
6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986
6987
6988
6989
6990
6991
6992
6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314
7315
7316
7317
7318
7319
7320
7321
7322
7323
7324
7325
7326
7327
7328
7329
7330
7331
7332
7333
7334
7335
7336
7337
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351
7352
7353
7354
7355
7356
7357
7358
7359
7360
7361
7362
7363
7364
7365
7366
7367
7368
7369
7370
7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7400
7401
7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422
7423
7424
7425
7426
7427
7428
7429
7430
7431
7432
7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469
7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7
```


M08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 103
T43 READ DATA WITH NON-EXISTENT MEMORY

SEQ 0103

```

5448
5449 015062 005737 001542      2$:  TST      T.WC      ;TEST IF WORD COUNT CORRECT
5450 015066 001411              BEQ      3$          ;YES - SKIP
5451 015070 012737 052707 001450  MOV     #EM10,EM10N ;SET MESSAGE
5452 015076 005037 001202              CLR     $REG10      ;SET VALUES
5453 015102 013737 001542 001204  MOV     T.WC,$REG11
5454 015110 104010              ERROR   10
5455
5456 015112      3$:
5457
5458  ;*****
5459  ;*TEST 44      UNIBUS PARITY ERROR
5460  ;*
5461  ;*      INITIALIZE A MEMORY LOCATION WITH BAD
5462  ;*      PARITY. ISSUE A WRITE DATA OF 400 WORDS
5463  ;*      STARTING AT A LOCATION 110 WORDS BEFORE
5464  ;*      THE LOCATION WITH BAD PARITY. MAKE SURE
5465  ;*      THAT UNIBUS PARITY ERROR SETS.
5466  ;*
5467  ;*      NOTE: THIS TEST IS ONLY EXECUTED IF
5468  ;*      MEMORY PARITY OPTION EXISTS FOR
5469  ;*      BUFFER.
5470  ;*
5471  ;*****
5472  ;*
5473  ;*
5474  ;*
5475  ;*
5476  ;*
5477  ;*
5478  ;*
5479  ;*
5480  ;*
5481  ;*
5482  ;*
5483  ;*
5484  ;*
5485  ;*
5486  ;*
5487  ;*
5488  ;*
5489  ;*
5490  ;*
5491  ;*
5492  ;*
5493  ;*
5494  ;*
5495  ;*
5496  ;*
5497  ;*
5498  ;*
5499  ;*
5500  ;*
5501  ;*
5502  ;*
5503  ;*

```


N08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 104
T44 UNIBUS PARITY ERROR

SEG 0104

5504	015236	012746	015244		MOV	#10\$,-(SP)	;PUT ADDRESS ON STACK
5505	015242	000002			RTI		;SET PRI
5506	015244			10\$:			
5507							
5508	015244	032737	000100	001656	BIT	#PARBK1,OPTFLG	;TEST IF PARITY BANK 1 AVAIL
5509	015252	001403			BEQ	4\$;NO - SKIP
5510	015254	012777	000004	164422	MOV	#BIT2,MMCSR2	;SET WRITE WRONG PARITY ENABLE
5511	015262	012777	000004	164412	MOV	#BIT2,MMCSR1	;SET WRITE WRONG PARITY BIT
5512	015270	012737	063032	063030	MOV	#OBUFF+226,OBUFF+224	
5513	015276	012737	063030	063026	MOV	#OBUFF+224,OBUFF+222	;WRITE ALL ONES INTO BUFFER W/BAD PARITY
5514	015304	012777	000001	164370	MOV	#BIT0,MMCSR1	;CLEAR CONTROL BIT, SET IE BIT
5515	015312	032737	000100	001656	BIT	#PARBK1,OPTFLG	;TEST IF BANK 1 AVAIL
5516	015320	001403			BEQ	5\$;NO SKIP
5517	015322	012777	000001	164354	MOV	#BIT0,MMCSR2	;CLEAR CONTROL BIT, SET IE
5518							
5519	015330	013746	001622		MOV	RKPRI,-(SP)	;SET OLD PRIORITY
5520	015334	012746	015342		MOV	#11\$,-(SP)	;SET ADDRESS
5521	015340	000002			RTI		;RESTORE PRI
5522	015342			11\$:			
5523							
5524	015342	104417			TLOADRK		;LOAD RK REGS
5525	015344	104430			TWAT96		;WAIT FOR INTERRUPT
5526	015346	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5527	015350	005077	164326		CLR	MMCSR1	;TURN OFF CSI
5528	015354	032737	000100	001656	BIT	#PARBK1,OPTFLC	;TEST IF BANK 1 PARITY PRESENT
5529	015362	001402			BEQ	6\$;NO SKIP
5530	015364	005077	164314		CLR	MMCSR2	;TURN OFF BANK 1
5531	015370	005037	063030		CLR	OBUFF+224	
5532	015374	005037	063026		CLR	OBUFF+222	;CLEAR BAD PARITY ERROR
5533							
5534	015400	004737	033704		JSR	PC,OPTTST	;RESET OPTIONS
5535							
5536	015404	104422			TCHKWE		;CHECK OPERATION WITH ERROR
5537	015406	000000			0		
5538	015410	000000			0		
5539	015412	000100			UPERR		;UNIBUS PARITY ERROR
5540	015414	104004			ERROR	4; OR 5,6,7	;REPORT ALL DISCREPANCIES
5541							
5542	015416	012737	047425	057662	MOV	#OPER44,DF010A	;SET MESSAGES FOR REPORT
5543	015424	012737	052734	001450	MOV	#EM11,EM10N	
5544	015432	023727	001544	063030	CMP	T.BA,#OBUFF+224	;CHECK IF BA IN RANGE
5545	015440	103010			BHIS	14\$;NOT TO LOW - SKIP
5546	015442	012737	063032	001202	MOV	#OBUFF+226,\$REG10	;SET VALUES FOR REPORT
5547	015450	013737	001544	001204	MOV	T.BA,\$REG11	
5548	015456	104010			ERROR	10	
5549	015460	000413			BR	16\$	
5550							
5551	015462	023727	001544	063034	14\$:	CMP	T.BA,#OBUFF+230 ;CHECK IF BA IN RANGE
5552	015470	101407			BLOS	16\$;YES - SKIP
5553	015472	012737	063034	001202	MOV	#OBUFF+230,\$REG10	;SET VALUES
5554	015500	012737	001544	001204	MOV	#T.BA,\$REG11	
5555	015506	104010			ERROR	10	
5556							
5557	015510	012737	052707	001450	16\$:	MOV	#EM10,EM10N ;SET MESSAGE
5558	015516	023727	001542	177512	CMP	T.WC,#-266	;CHECK IF WORD COUNT WITHIN RANGE
5559	015524	103007			BHIS	20\$;YES - SKIP


```

5560 015526 012737 177512 001202      MOV      #-266,$REG10      ;SET VALUES
5561 015534 013737 001542 001204      MOV      T.WC,$REG11
5562 015542 104010                      ERROR    10
5563
5564 015544 023727 001542 177513 20$:  CMP      T.WC,#-265      ;STILL CHECKING IF WC IN RANGE
5565 015552 101407                      BLOS    2$              ;YES - SKIP
5566 015554 012737 177513 001202      MOV      #-265,$REG10      ;SET VALUES
5567 015562 013737 001542 001204      MOV      T.WC,$REG11
5568 015570 104010                      ERROR    10
5569
5570 015572                      2$:
5571
5572                      .SBTTL  **MULTIPLE SECTOR OPERATIONS
5573
5574                      ::*****
5575                      ::*TEST 45      TWO SECTOR WRITE DATA (PART 1)
5576                      ::*
5577                      ::*      ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5578                      ::*      TRACK 0, SECTOR 0.  READ DATA BACK ONE SECTOR
5579                      ::*      AT A TIME AND MAKE SURE IT IS CORRECT.
5580                      ::*
5581                      ::*****
5582 015572 000004                      †ST45:  SCOPE
5583 015574 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
5584 015602 104416                      TSSINIT                      ;CLEAR SUBSYSTEM
5585 015604 104003                      ERROR    3                  ;BAD INIT ERROR
5586
5587 015606 004437 034574                      JSR      R4,LRLOAD          ;LOAD "L" REGS
5588 015612 000123                      WRDATA                      ;WRDATA
5589 015614 177000                      -1000                        ;-1000 WORDS
5590 015616 062604                      OBUFF                        ;OBUFF IS BUFF ADDRESS
5591 015620 000                      .BYTE    0                    ;SECTOR 0
5592 015621 000                      .BYTE    0                    ;TRACK 0
5593 015622 000312                      312                          ;CYLINDER 312
5594
5595 015624 004437 040776                      JSR      R4,GENCOM          ;GENERATE DATA
5596 015630 000015                      15                            ;PATTERN 15
5597 015632 001000                      1000                         ;1000 WORDS
5598
5599 015634 104417                      TLOADRK                      ;LOAD RK REGS
5600 015636 104430                      TWAT96                        ;WAIT FOR INTERRUPT
5601 015640 104002                      ERROR    2                    ;TO SLOW/NOT COMPLETE ERROR
5602
5603 015642 104421                      TCHKOP                      ;CHECK OPERATION FOR ANY ERRORS
5604 015644 104004                      ERROR    4 ;OR 5, 6, 7. 10 ;REPORT ALL ERRORS
5605
5606 015646 004437 040776                      JSR      R4,GENCOM          ;CLEAR Ibuff
5607 015652 002007                      2007                          ;TO ALL 1'S
5608 015654 001000                      1000
5609
5610 015656 004437 034574                      JSR      R4,LRLOAD          ;LOAD "L" REGS
5611 015662 000121                      RDDATA                      ;RDDATA
5612 015664 177400                      -400                         ;-400 WORDS
5613 015666 060604                      Ibuff                        ;IBUFF IS BUFF ADDRESS
5614 015670 000                      .BYTE    0                    ;SECTOR 0
5615 015671 000                      .BYTE    0                    ;TRACK 0

```



```

5616 015672 000312          312          ;CYLINDER 312
5617
5618 015674 104417          TLOADRK          ;LOAD RK REGS
5619 015676 104424          TWAT32           ;WAIT FOR INTERRUPT
5620 015700 104002          ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
5621
5622 015702 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5623 015704 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5624
5625 015706 004437 034574    JSR R4,LRLOAD   ;LOAD "L" REGS
5626 015712 000121          RDDATA          ;RDDATA
5627 015714 177400          -400            ;-400 WORDS
5628 015716 061604          Ibuff+1000     ;IBUFF+1000 IS BUFF ADDRESS
5629 015720 001           .BYTE 1         ;SECTOR 1
5630 015721 000           .BYTE 0         ;TRACK 0
5631 015722 000312          312            ;CYLINDER 312
5632
5633 015724 104417          TLOADRK          ;LOAD RK REGS
5634 015726 104424          TWAT32           ;WAIT FOR INTERRUPT
5635 015730 104002          ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
5636
5637 015732 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
5638 015734 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5639
5640 015736 004437 040776    JSR R4,GENCOM   ;COMPARE DATA
5641 015742 100000          100000         ;
5642 015744 001000          1000           ;1000 WORDS
5643 015746 000413          BR 2$          ;NO MISCOMPARES-EXIT
5644 015750 104015          ERROR 15       ;REPORT FIRST ERROR
5645
5646 015752 013700 001634    MOV ERRLMT,RO   ;GET ERROR LIMIT
5647 015756 005300          64$: DEC RO      ;DECREMENT COUNT
5648 015760 001406          BEQ 65$        ;IF ZERO - EXIT
5649 015762 004437 040776    JSR R4,GENCOM   ;CONTINUE DATA COMPARE
5650 015766 040000          40000         ;
5651 015770 000402          BR 65$        ;NO MORE ERRORS - EXIT
5652 015772 104016          ERROR 16       ;REPORT NEXT ERROR
5653 015774 000770          BR 64$        ;LOOP
5654 015776
5655
5656 015776
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666 015776 000004          TST46: SCOPE
5667 016000 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
5668 016006 104416          TSSINIT        ;CLEAR SUBSYSTEM
5669 016010 104003          ERROR 3        ;BAD INIT ERROR
5670
5671 016012 004437 034574    JSR R4,LRLOAD   ;LOAD "L" REGS

```

```

*****
;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.
*****

```


5672	016016	000123		WRDATA		:WRDATA
5673	016020	177000		-1000		: -1000 WORDS
5674	016022	062604		OBUFF		: OBUFF IS BUFF ADDRESS
5675	016024	023		.BYTE	23	: SECTOR 23
5676	016025	000		.BYTE	0	: TRACK 0
5677	016026	000312		312		: CYLINDER 312
5678						
5679	016030	004437	040776	JSR	R4,GENCOM	: GENERATE DATA
5680	016034	000016		16		: PATTERN 16
5681	016036	001000		1000		: 1000 WORDS
5682						
5683	016040	104417		TLOADRK		: LOAD RK REGS
5684	016042	104430		TWAT96		: WAIT FOR INTERRUPT
5685	016044	104002		ERROR	2	: TO SLOW/NOT COMPLETE ERROR
5686						
5687	016046	104421		TCHKOP		: CHECK OPERATION FOR ANY ERRORS
5688	016050	104004		ERROR	4 ;OR 5, 6, 7, 10	: REPORT ALL ERRORS
5689						
5690						
5691						
5692						
5693						
5694	016052	004437	034574	JSR	R4,LRLoad	: LOAD "L" REGS
5695	016056	000121		RDDATA		: RDDATA
5696	016060	177400		-400		: -400 WORDS
5697	016062	060604		IBUFF		: IBUFF IS BUFF ADDRESS
5698	016064	023		.BYTE	23	: SECTOR 23
5699	016065	000		.BYTE	0	: TRACK 0
5700	016066	000312		312		: CYLINDER 312
5701						
5702	016070	004437	040776	JSR	R4,GENCOM	: CLEAR IBUFF TO ALL ONES
5703	016074	002007		2007		
5704	016076	001000		1000		
5705						
5706	016100	104417		TLOADRK		: LOAD RK REGS
5707	016102	104424		TWAT32		: WAIT FOR INTERRUPT
5708	016104	104002		ERROR	2	: TO SLOW/NOT COMPLETE ERROR
5709						
5710	016106	104421		TCHKOP		: CHECK OPERATION FOR ANY ERRORS
5711	016110	104004		ERROR	4 ;OR 5, 6, 7, 10	: REPORT ALL ERRORS
5712						
5713	016112	004437	034574	JSR	R4,LRLoad	: LOAD "L" REGS
5714	016116	000121		RDDATA		: RDDATA
5715	016120	177400		-400		: -400 WORDS
5716	016122	061604		IBUFF+1000		: IBUFF+1000 IS BUFF ADDRESS
5717	016124	024		.BYTE	24	: SECTOR 24
5718	016125	000		.BYTE	0	: TRACK 0
5719	016126	000312		312		: CYLINDER 312
5720						
5721	016130	104417		TLOADRK		: LOAD RK REGS
5722	016132	104424		TWAT32		: WAIT FOR INTERRUPT
5723	016134	104002		ERROR	2	: TO SLOW/NOT COMPLETE ERROR
5724						
5725	016136	104421		TCHKOP		: CHECK OPERATION FOR ANY ERRORS
5726	016140	104004		ERROR	4 ;OR 5, 6, 7, 10	: REPORT ALL ERRORS
5727						

E09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 108
T46 TWO SECTOR WRITE DATA (PART 2)

SEQ 0108

```

5728 016142 004437 040776      JSR      R4,GENCOM      ;COMPARE DATA
5729 016146 100000              100000
5730 016150 001000              1000      ;1000 WORDS
5731 016152 000413              BR       15             ;NO ERRORS-SKIP
5732 016154 104015              ERROR    15             ;REPORT FIRST ERROR
5733
5734 016156 013700 001634          MOV      ERRLMT,RO      ;GET ERROR LIMIT
5735 016162 005300      64$:    DEC      RO          ;DECREMENT COUNT
5736 016164 001406              BEQ      65$            ;IF ZERO - EXIT
5737 016166 004437 040776      JSR      R4,GENCOM      ;CONTINUE DATA COMPARE
5738 016172 040000              40000
5739 016174 000402              BR       65$            ;NO MORE ERRORS - EXIT
5740 016176 104016              ERROR    16             ;REPORT NEXT ERROR
5741 016200 000770              BR       64$            ;LOOP
5742 016202
5743 016202
5744
5745
5746
5747
5748
5749
5750
5751
5752 016202 000004      *ST47: SCOPE
5753 016204 012737 000062 001262  MOV      #50.,$TIMES    ;:DO 50. ITERATIONS
5754 016212 104416              TSSINIT                ;:CLEAR SUBSYSTEM
5755 016214 104003              ERROR    3              ;:BAD INIT ERROR
5756
5757 016216 004437 034574      JSR      R4,LRLOAD      ;:LOAD "L" REGS
5758 016222 000123              WRDATA                ;:WRDATA
5759 016224 177377              -401                  ;:-401 WORDS
5760 016226 062604              OBUFF                 ;:OBUFF IS BUFF ADDRESS
5761 016230 010                      .BYTE    10            ;:SECTOR 10
5762 016231 000                      .BYTE    0              ;:TRACK 0
5763 016232 000312              312                   ;:CYLINDER 312
5764
5765 016234 004437 040776      JSR      R4,GENCOM      ;:GENERATE DATA
5766 016240 000002              2                      ;:PATTERN 2
5767 016242 001000              1000                   ;:1000 WORDS
5768
5769 016244 104417              TLOADRK                ;:LOAD RK REGS
5770 016246 104430              TWAT96                 ;:WAIT FOR INTERRUPT
5771 016250 104002              ERROR    2              ;:TO SLOW/NOT COMPLETE ERROR
5772
5773 016252 104421              TCHKOP                 ;:CHECK OPERATION FOR ANY ERRORS
5774 016254 104004              ERROR    4 ;OR 5, 6, 7 ;:REPORT ALL ERRORS
5775 ; CLEAR LAST 377 WORDS OF OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
5776 016256 012700 063606      MOV      #OBUFF+1002,RO ;:GET STARTING ADDRESS TO BE CLEARED
5777 016262 012701 000377      MOV      #377,R1        ;:NUMBER OF WORDS
5778 016266 005020      1$:    CLR      (RO)+         ;:CLEAR WORD
5779 016270 005301              DEC      R1             ;:DEC COUNT
5780 016272 001375              BNE     1$              ;:LOOP UNTIL COUNT ZERO
5781 016274 004437 040776      JSR      R4,GENCOM      ;:CLEAR IBUFF TO ONES
5782 016300 002007              2007
5783 016302 001000              1000

```


F09

```

5784
5785 016304 004437 034574      JSR      R4,LRLOAD      ;LOAD "L" REGS
5786 016310 000121      RDDATA      ;RDDATA
5787 016312 177400      -400      ;-400 WORDS
5788 016314 060604      Ibuff      ;IBUFF IS BUFF ADDRESS
5789 016316      010      .BYTE 10      ;SECTOR 10
5790 016317      000      .BYTE 0      ;TRACK 0
5791 016320 000312      312      ;CYLINDER 312
5792
5793 016322 104417      TLOADRK     ;LOAD RK REGS
5794 016324 104424      TWAT32     ;WAIT FOR INTERRUPT
5795 016326 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
5796
5797 016330 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
5798 016332 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5799 016334 004437 034574      JSR      R4,LRLOAD     ;LOAD "L" REGS
5800 016340 000121      RDDATA     ;RDDATA
5801 016342 177400      -400      ;-400 WORDS
5802 016344 061604      Ibuff+1000 ;IBUFF+1000 IS BUFF ADDRESS
5803 016346      011      .BYTE 11     ;SECTOR 11
5804 016347      000      .BYTE 0      ;TRACK 0
5805 016350 000312      312      ;CYLINDER 312
5806
5807 016352 104417      TLOADRK     ;LOAD RK REGS
5808 016354 104424      TWAT32     ;WAIT FOR INTERRUPT
5809 016356 104002      ERROR 2    ;TO SLOW/NOT COMPLETE ERROR
5810
5811 016360 004437 040776      JSR      R4,GENCOM     ;DATA COMPARE
5812 016364 100000      100000
5813 016366 001000      1000      ;1000 WORDS
5814 016370 000413      BR 25     ;NO ERROR-SKIP
5815 016372 104015      ERROR 15  ;REPORT FIRST ERROR
5816
5817 016374 013700 001634      MOV      ERLMT,RO     ;GET ERROR LIMIT
5818 016400 005300      64$: DEC RO      ;DECREMENT COUNT
5819 016402 001406      BEQ 65$  ;IF ZERO - EXIT
5820 016404 004437 040776      JSR      R4,GENCOM     ;CONTINUE DATA COMPARE
5821 016410 040000      40000
5822 016412 000402      BR 65$   ;NO MORE ERRORS - EXIT
5823 016414 104016      EROR 16  ;REPORT NEXT ERROR
5824 016416 000770      BR 64$   ;LOOP
5825 016420
5826
5827 016420      2$:
5828 ;*****
5829 ;*TEST 50 MID-TRANSFER SEEK ON WRITE (PART 1)
5830 ;*
5831 ;* ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5832 ;* TRACK 0, SECTOR 25. READ DATA BACK ONE SECTOR
5833 ;* AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5834 ;* DID TAKE PLACE.
5835 ;*
5836 ;*****
5837 016420 000004      TST50: SCOPE
5838 016422 012737 000062 001262      MOV      #50.,$TIMES ;DO 50. ITERATIONS
5839 016430 104416      TSSINIT ;CLEAR SUBSYSTEM
  
```



```

5840 016432 104003          ERROR 3          ;BAD INIT ERROR
5841
5842 016434 004437 034574  JSR    R4,LRLOAD  ;LOAD "L" REGS
5843 016440 000123          WRDATA          ;WRDATA
5844 016442 177000          -1000          ;-1000 WORDS
5845 016444 062604          OBUFF          ;OBUFF IS BUFF ADDRESS
5846 016446      025        .BYTE 25        ;SECTOR 25
5847 016447      000        .BYTE 0          ;TRACK 0
5848 016450 000312          312           ;CYLINDER 312
5849
5850 016452 004437 040776  JSR    R4,GENCOM ;GENERATE DATA
5851 016456 000003          3             ;PATTERN 3
5852 016460 001000          1000          ;1000 WORDS
5853
5854 016462 104417          TLOADRK        ;LOAD RK REGS
5855 016464 104430          TWAT96        ;WAIT FOR INTERRUPT
5856 016466 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
5857
5858 016470 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5859 016472 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5860          :          A TRACK ERROR PRINTED OUT AT THE END OF THE OPERATION INDICATES A
5861          :          MID-TRANSFER HEAD SWITCH DID NOT OCCUR.
5862 016474 004437 040776  JSR    R4,GENCOM
5863 016500 002007          2007
5864 016502 001000          1000
5865
5866 016504 004437 034574  JSR    R4,LRLOAD  ;LOAD "L" REGS
5867 016510 000121          RDDATA          ;RDDATA
5868 016512 177400          -400           ;-400 WORDS
5869 016514 060604          IBUFF          ;IBUFF IS BUFF ADDRESS
5870 016516      025        .BYTE 25        ;SECTOR 25
5871 016517      000        .BYTE 0          ;TRACK 0
5872 016520 000312          312           ;CYLINDER 312
5873
5874 016522 104417          TLOADRK        ;LOAD RK REGS
5875 016524 104425          TWAT48        ;WAIT FOR INTERRUPT
5876 016526 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
5877
5878 016530 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5879 016532 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5880
5881 016534 004437 034574  JSR    R4,LRLOAD  ;LOAD "L" REGS
5882 016540 000121          RDDATA          ;RDDATA
5883 016542 177400          -400           ;-400 WORDS
5884 016544 061604          IBUFF+1000    ;IBUFF+1000 IS BUFF ADDRESS
5885 016546      000        .BYTE 0          ;SECTOR 0
5886 016547      001        .BYTE 1          ;TRACK 1
5887 016550 000312          312           ;CYLINDER 312
5888
5889 016552 104417          TLOADRK        ;LOAD RK REGS
5890 016554 104425          TWAT48        ;WAIT FOR INTERRUPT
5891 016556 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
5892
5893 016560 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5894 016562 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5895

```



```

5896 016564 004437 040776 JSR R4,GENCOM ;COMPARE DATA
5897 016570 100000 100000 ;1000 WORDS
5898 016572 001000 1000 ;NO ERRORS-SKIP
5899 016574 000413 BR 1$ ;REPORT FIRST ERROR
5900 016576 104015 ERROR 15
5901
5902 016600 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
5903 016604 005300 64$: DEC RO ;DECREMENT COUNT
5904 016606 001406 BEQ 65$ ;IF ZERO - EXIT
5905 016610 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5906 016614 040000 40000
5907 016616 000402 BR 65$ ;NO MORE ERRORS - EXIT
5908 016620 104016 ERROR 16 ;REPORT NEXT ERROR
5909 016622 000770 BR 64$ ;LOOP
5910
5911 65$:
5912 016624 1$:
5913 *****
5914 *TEST 51 MID-TRANSFER SEEK ON WRITE (PART 2)
5915 *
5916 * ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5917 * TRACK 2, SECTOR 25. READ DATA BACK ONE SECTOR
5918 * AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5919 * DID TAKE PLACE.
5920 *
5921 *****
5922 016624 000004 TST51: SCOPE
5923 016626 012737 000062 001262 MOV #50.,$TIMES ;;DO 50. ITERATIONS
5924 016634 104416 TSSINIT ;CLEAR SUBSYSTEM
5925 016636 104003 ERROR 3 ;BAD INIT ERROR
5926
5927 016640 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
5928 016644 000123 WRDATA ;WRDATA
5929 016646 177000 -1000 ;-1000 WORDS
5930 016650 062604 OBUFF ;OBUFF IS BUFF ADDRESS
5931 016652 025 .BYTE 25 ;SECTOR 25
5932 016653 002 .BYTE 2 ;TRACK 2
5933 016654 000312 312 ;CYLINDER 312
5934
5935 016656 004437 040776 JSR R4,GENCOM ;GENERATE DATA
5936 016662 000004 4 ;PATTERN 4
5937 016664 001000 1000 ;1000 WORDS
5938
5939 016666 104417 TLOADRK ;LOAD RK REGS
5940 016670 104430 TWAT96 ;WAIT FOR INTERRUPT
5941 016672 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5942 ; A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A
5943 ; MID-TRANSFER SEEK DID NOT OCCUR.
5944 016674 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5945 016676 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5946
5947 016700 004437 040776 JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
5948 016704 002007 2007
5949 016706 001000 1000
5950
5951 016710 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS

```



```

5952 016714 000121          RDDATA          ;RDDATA
5953 016716 177400          -400           ; -400 WORDS
5954 016720 060604          Ibuff          ;IBUFF IS BUFF ADDRESS
5955 016722      025        .BYTE 25       ;SECTOR 25
5956 016723      002        .BYTE 2           ;TRACK 2
5957 016724 000312          312           ;CYLINDER 312
5958
5959 016726 104417          TLOADRK        ;LOAD RK REGS
5960 016730 104425          TWAT48        ;WAIT FOR INTERRUPT
5961 016732 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
5962
5963 016734 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5964 016736 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5965
5966 016740 004437 034574    JSR R4,LRLOAD  ;LOAD "L" REGS
5967 016744 000121          RDDATA          ;RDDATA
5968 016746 177400          -400           ; -400 WORDS
5969 016750 061604          Ibuff+1000    ;IBUFF+1000 IS BUFF ADDRESS
5970 016752      000        .BYTE 0         ;SECTOR 0
5971 016753      000        .BYTE 0         ;TRACK 0
5972 016754 000313          313           ;CYLINDER 313
5973
5974 016756 104417          TLOADRK        ;LOAD RK REGS
5975 016760 104425          TWAT48        ;WAIT FOR INTERRUPT
5976 016762 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
5977
5978 016764 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
5979 016766 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5980
5981 016770 004437 040776    JSR R4,GENCOM  ;COMPARE DATA
5982 016774 100000          100000        ;1000 WORDS
5983 016776 001000          1000          ;NO MISCOMPARES-SKIP
5984 017000 000413          BR 15         ;REPORT 1ST ERROR
5985 017002 104015          ERROR 15
5986
5987 017004 013700 001634    MOV ERRLMT,RO  ;GET ERROR LIMIT
5988 017010 005300          DEC RO        ;DECREMENT COUNT
5989 017012 001406          BEQ 65$      ;IF ZERO - EXIT
5990 017014 004437 040776    JSR R4,GENCOM  ;CONTINUE DATA COMPARE
5991 017020 040000          40000        ;NO MORE ERRORS - EXIT
5992 017022 000402          BR 65$      ;REPORT NEXT ERROR
5993 017024 104016          ERROR 16
5994 017026 000770          BR 64$      ;LOOP
5995 017030
5996
5997 017030
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007

```

64\$:

65\$:

1\$:

```

*****
*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.
*

```



```

6008
6009 017030 000004
6010 017032 012737 000062 001262
6011 017040 104416
6012 017042 104003
6013
6014
6015
6016
6017 017044 004437 040776
6018 017050 000015
6019 017052 001000
6020
6021 017054 004437 040776
6022 017060 002007
6023 017062 001000
6024
6025 017064 004437 034574
6026 017070 000121
6027 017072 177000
6028 017074 060604
6029 017076 000
6030 017077 000
6031 017100 000312
6032
6033 017102 104417
6034 017104 104430
6035 017106 104002
6036
6037 017110 104421
6038 017112 104004
6039
6040 017114 004437 040776
6041 017120 100000
6042 017122 001000
6043 017124 000413
6044 017126 104015
6045
6046 017130 013700 001634
6047 017134 005300
6048 017136 001406
6049 017140 004437 040776
6050 017144 040000
6051 017146 000402
6052 017150 104016
6053 017152 000770
6054 017154
6055
6056 017154
6057
6058
6059
6060
6061
6062
6063

```

```

*****
TST52: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA (PART 1)

; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA PART 1
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
RDDATA ;RDDATA
-1000 ;-1000 WORDS
IBUFF ;IBUFF 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,GENCOM ;COMPARE DATA
100000
1000 ;1000 WORDS
BR 1$ ;NO MISCOMPARES-SKIP
ERROR 15

MOV ERLMT,R0 ;GET ERROR LIMIT
DEC R0 ;DECREMENT COUNT
BEQ 65$ ;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 65$ ;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 64$ ;LOOP

64$:
65$:
1$:
*****
*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.
*

```



```

6064
6065
6066
6067
6068 017154 000004
6069 017156 012737 000062 001262
6070 017164 104416
6071 017166 104003
6072
6073
6074
6075 017170 004437 040776
6076 017174 000016
6077 017176 001000
6078
6079 017200 004437 040776
6080 017204 002007
6081 017206 001000
6082 017210 004437 034574
6083 017214 000121
6084 017216 177000
6085 017220 060604
6086 017222 023
6087 017223 000
6088 017224 000312
6089
6090 017226 104417
6091 017230 104430
6092 017232 104002
6093
6094 017234 104421
6095 017236 104004
6096
6097 017240 004437 040776
6098 017244 100000
6099 017246 001000
6100 017250 000413
6101 017252 104015
6102
6103 017254 013700 001634
6104 017260 005300
6105 017262 001406
6106 017264 004437 040776
6107 017270 040000
6108 017272 000402
6109 017274 104016
6110 017276 000770
6111 017300
6112
6113 017300
6114
6115
6116
6117
6118
6119

```

```

;* NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE
;* EXECUTED BEFORE THIS TEST.
*****
TST53: SCOPE
MOV #50.,$TIMES ;:DO 50. ITERATIONS
TSSINIT ;:CLEAR SUBSYSTEM
ERROR 3 ;:BAD INIT ERROR

;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)

JSR R4,GENCOM ;:GENERATE DATA
16 ;:PATTERN 16
1000 ;:1000 WORDS

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

JSR R4,LRLOAD ;:LOAD "L" REGS
RDDATA ;:RDDATA
-1000 ;:-1000 WORDS
IBUFF ;:IBUFF 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,GENCOM ;:COMPARE DATA
100000
1000 ;:1000 WORDS
BR 1$ ;:NO MISCOMPARES-SKIP
ERROR 15 ;:REPORT 1ST ERROR

MOV ERRLMT,R0 ;:GET ERROR LIMIT
DEC R0 ;:DECREMENT COUNT
BEQ 65$ ;:IF ZERO - EXIT
JSR R4,GENCOM ;:CONTINUE DATA COMPARE
40000
BR 65$ ;:NO MORE ERRORS - EXIT
ERROR 16 ;:REPORT NEXT ERROR
BR 64$ ;:LOOP

64$:
65$:
1$:
*****
*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.
;:

```



```

6120
6121
6122
6123
6124
6125 017300 000004
6126 017302 012737 000062 001262
6127 017310 104416
6128 017312 104003
6129
6130
6131
6132 017314 004437 040776
6133 017320 000002
6134 017322 000401
6135
6136 017324 004437 040776
6137 017330 002007
6138 017332 001000
6139
6140 017334 004437 034574
6141 017340 000121
6142 017342 177377
6143 017344 060604
6144 017346 010
6145 017347 000
6146 017350 000312
6147
6148 017352 104417
6149 017354 104430
6150 017356 104002
6151
6152 017360 104421
6153 017362 104004
6154
6155 017364 004437 040776
6156 017370 100000
6157 017372 000401
6158 017374 000413
6159 017376 104015
6160
6161 017400 013700 001634
6162 017404 005300
6163 017406 001406
6164 017410 004437 040776
6165 017414 040000
6166 017416 000402
6167 017420 104016
6168 017422 000770
6169 017424
6170 017424
6171
6172
6173
6174
6175

```

```

;*
;* NOTE: TWO SECTOR WRITE DATA (PART 3) MUST BE
;* EXECUTED BEFORE THIS TEST.
;*
*****
TST54: SCOPE
MOV #50.,$TIMES ;DO 50. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)

JSR R4,GENCOM ;GENERATE DATA
2 ;PATTERN 2
401 ;401 WORDS

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

JSR R4,LRLOAD ;LOAD "L" REGS
RDDATA ;RDDATA
-401 ;-401 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 10 ;SECTOR 10
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

TLOADRK ;LOAD RK REGS
TWT96 ;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
401 ;401 WORDS
BR 1$ ;NO MISCOMPARES-SKIP
ERROR 15 ;PRINT FIRST ERROR

MOV ERRLMT,R0 ;GET ERROR LIMIT
DEC R0 ;DECREMENT COUNT
BEQ 65$ ;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 65$ ;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 64$ ;LOOP

64$:
65$:
1$:
*****
;*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

```


M09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 116
T55 MID-TRANSFER SEEK ON READ (PART 1)

SEQ 0116

```

6176 ;* READ AND A MID-TRANSFER SEEK DOES OCCUR.
6177 ;*
6178 ;* NOTE: MID-TRANSFER SEEK ON WRITE (PART 1) MUST BE
6179 ;* EXECUTED BEFORE THIS TEST.
6180 ;*
6181 ;* *****
6182 017424 000004 TST55: SCOPE
6183 017426 012737 000062 001262 MOV #50.,$TIMES ;DO 50. ITERATIONS
6184 017434 104416 TSSINIT ;CLEAR SUBSYSTEM
6185 017436 104003 ERROR 3 ;BAD INIT ERROR
6186
6187 ; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
6188 017440 004437 040776 JSR R4,GENCOM ;GENERATE DATA
6189 017444 000003 3 ;PATTERN 3
6190 017446 001000 1000 ;1000 WORDS
6191
6192 017450 004437 040776 JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6193 017454 002007 2007
6194 017456 001000 1000
6195
6196 017460 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
6197 017464 000121 RDDATA ;RDDATA
6198 017466 177000 -1000 ;-1000 WORDS
6199 017470 060604 Ibuff ;IBUFF IS BUFF ADDRESS
6200 017472 025 .BYTE 25 ;SECTOR 25
6201 017473 000 .BYTE 0 ;TRACK 0
6202 017474 000312 312 ;CYLINDER 312
6203
6204 017476 104417 TLOADRK ;LOAD RK REGS
6205 017500 104430 TWAT96 ;WAIT FOR INTERRUPT
6206 017502 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6207
6208 017504 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6209 017506 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6210
6211 017510 004437 040776 JSR R4,GENCOM ;COMPARE DATA
6212 017514 100000 100000
6213 017516 001000 1000 ;1000 WORDS
6214 017520 000413 BR 1$ ;NO MISCOMPARES-SKIP
6215 017522 104015 ERROR 15 ;PRINT FIRST ERROR
6216
6217 017524 013700 001634 MOV ERRLMT,RO ;GET ERROR LIMIT
6218 017530 005300 64$: DEC RO ;DECREMENT COUNT
6219 017532 001406 BEQ 65$ ;IF ZERO - EXIT
6220 017534 004437 040776 JSR R4,GENCOM ;CONTINUE DATA COMPARE
6221 017540 040000 40000
6222 017542 000402 BR 65$ ;NO MORE ERRORS - EXIT
6223 017544 104016 ERROR 16 ;REPORT NEXT ERROR
6224 017546 000770 BR 64$ ;LOOP
6225 017550
6226 017550
6227 ;* *****
6228 ;* TEST 56 MID-TRANSFER SEEK ON READ (PART 2)
6229 ;*
6230 ;* ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6231 ;* TRACK 2, SECTOR 25. VERIFY THAT CORRECT DATA IS

```



```

6232
6233
6234
6235
6236
6237
6238 017550 000004
6239 017552 012737 000062 001262
6240 017560 104416
6241 017562 104003
6242
6243
6244 017564 004437 040776
6245 017570 000004
6246 017572 001000
6247
6248 017574 004437 040776
6249 017600 002007
6250 017602 001000
6251
6252 017604 004437 034574
6253 017610 000121
6254 017612 177000
6255 017614 060604
6256 017616 025
6257 017617 002
6258 017620 000312
6259
6260 017622 104417
6261 017624 104430
6262 017626 104002
6263
6264 017630 104421
6265 017632 104004
6266
6267 017634 004437 040776
6268 017640 100000
6269 017642 001000
6270 017644 000413
6271 017646 104015
6272
6273 017650 013700 001634
6274 017654 005300
6275 017656 001406
6276 017660 004437 040776
6277 017664 040000
6278 017666 000402
6279 017670 104016
6280 017672 000770
6281 017674
6282 017674
6283
6284
6285
6286
6287

```

```

;* READ AND A MID-TRANSFER SEEK DOES OCCUR.
;*
;* NOTE: MID-TRANSFER SEEK ON WRITE (PART 2) MUST BE
;* EXECUTED BEFORE THIS TEST.
;*****
;T56: SCOPE
MOV #50.,$TIMES ;:DO 50. ITERATIONS
TSSINIT ;:CLEAR SUBSYSTEM
ERROR 3 ;:BAD INIT ERROR
;
; GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)
JSR R4,GENCOM ;:GENERATE DATA
4 ;:PATTERN 4
1000 ;:1000 WORDS
;
JSR R4,GENCOM ;:CLEAR Ibuff TO ALL ONES
2007
1000
;
JSR R4,LRLOAD ;:LOAD "L" REGS
RDDATA ;:RDDATA
-1000 ;:-1000 WORDS
IBUFF ;:IBUFF IS BUFF ADDRESS
.BYTE 25 ;:SECTOR 25
.BYTE 2 ;:TRACK 2
312 ;:CYLINDER 312
;
TLOADRK ;:LOAD RK REGS
TWT96 ;:WAIT FOR INTERRUPT
ERROR 2 ;:TO SLOW/NOT COMPLETE ERROR
;
TCHKOP ;:CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7, 10 ;:REPORT ALL ERRORS
;
JSR R4,GENCOM ;:COMPARE DATA
100000
1000 ;:1000 WORDS
BR 1$ ;:NO MISCOMPARES-SKIP
ERROR 15 ;:REPORT FIRST ERROR
;
MOV ERRLMT,R0 ;:GET ERROR LIMIT
64$: DEC R0 ;:DECREMENT COUNT
BEQ 65$ ;:IF ZERO - EXIT
JSR R4,GENCOM ;:CONTINUE DATA COMPARE
40000
BR 65$ ;:NO MORE ERRORS - EXIT
ERROR 16 ;:REPORT NEXT ERROR
BR 64$ ;:LOOP
;
65$:
1$:
;*****
;*TEST 57 CYLINDER ADDRESS OVERFLOW (PART 1)
;
;
; ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,
; TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS

```


B10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 118
TS7 CYLINDER ADDRESS OVERFLOW (PART 1)

SEQ 0118

6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343

017674 000004
017676 012737 000062 001262
017704 104416
017706 104003

017710 004437 034574
017714 000121
017716 177400
017720 060604
017722 025
017723 002
017724 000632

017726 104417
017730 104434
017732 104002

017734 104421
017736 104004

017740 000004
017742 012737 000062 001262
017750 104416
017752 104003

017754 004437 034574
017760 000121
017762 177377
017764 060604
017766 025
017767 002
017770 000632

017772 104417
017774 104434
017776 104002

020000 104422
020002 001000
020004 000000
020006 000000
020010 104004

```

;* OVERFLOW ERROR DOES NOT OCCUR.
;*
*****
TEST7: SCOPE
MOV #50.,$TIMES ;;DO 50. ITERATIONS
TSSINIT ;;CLEAR SUBSYSTEM
ERROR 3 ;;BAD INIT ERROR

JSR R4,LRLOAD ;;LOAD "L" REGS
RDDATA ;;RDDATA
-400 ;;-400 WORDS
IBUFF ;;IBUFF IS BUFF ADDRESS
.BYTE 25 ;;SECTOR 25
.BYTE 2 ;;TRACK 2
632 ;;CYLINDER 632

TLOADRK ;;LOAD RK REGS
TWAT159 ;;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
*****
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.
*****
TEST60: SCOPE
MOV #50.,$TIMES ;;DO 50. ITERATIONS
TSSINIT ;;CLEAR SUBSYSTEM
ERROR 3 ;;BAD INIT ERROR

JSR R4,LRLOAD ;;LOAD "L" REGS
RDDATA ;;RDDATA
-401 ;;-401 WORDS
IBUFF ;;IBUFF IS BUFF ADDRESS
.BYTE 25 ;;SECTOR 25
.BYTE 2 ;;TRACK 2
632 ;;CYLINDER 632

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

C10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 119
T61 FORMAT IN 24 SECTOR FORMAT

SEQ 0119

6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399

```

020012 000004
020014 012737 000062 001262
020022 012737 000312 001664
020030 012737 020040 001110
020036 005001
020040
020040 104416
020042 104003
020044 012737 010127 001600
020052 013737 001626 001610
020060 012737 000074 001602
020066 110137 001607
020072 012737 062604 001604
020100 012737 000312 001614

020106 004437 040776
020112 001200

020114 104417
020116 104434
020120 104002

020122 104421
020124 104004

020126 104415

020130 005701
020132 001002
020134 005201
020136 000740
020140 012737 020150 001110 2$:
020146 005001
020150
020150 104416 3$:
020152 104003
020154 012737 010125 001600
020162 013737 001626 001610
020170 110137 001607
020174 012737 000312 001614

020202 004437 035114
020206 104421
020210 104004
020212 104013
020214 104002
    
```

```

*****
*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.
*****
T61:  SCOPE
      MOV     #50, $TIMES      ;DO 50. ITERATIONS
      MOV     #312, REFMT     ;SET REFORMAT SWITCH
      MOV     #1$, $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
      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

      JSR     R4, GENCOM      ;GENERATE HEADER
      1200    ;INCLUDE BAD SECTOR BITS

      TLOADRK                ;LOAD RK REGS
      TWAT159                ;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

      SCOP1                   ;LOCAL LOOP ON ERROR TO 1$

      TST     R1              ;R1 POINTING TO TRACK 0
      BNE     2$              ;NO-SKIP
      INC     R1              ;BUMP TO TRACK 1
      BR      1$              ;LOOP
      MOV     #3$, $LPERR     ;SET LOCAL LOOP ON ERROR
      CLR     R1              ;CLEAR TRACK POINTER

2$:   TSSINIT                ;CLEAR SUBSYSTEM
      ERROR   3              ;BAD INIT ERROR
      MOV     #RDHEAD!CFMT, L.CS1 ;LOAD READ 24(8) SECTOR FORMAT
      MOV     DRVNUM, L.CS2   ;LOAD DRIVE NUMBER
      MOV     R1, L.DT        ;LOAD TRACK
      MOV     #312, L.DCYL    ;LOAD CYLINDER

      JSR     R4, RDSTHD      ;GO READ STANDARD HEADER
      TCHKOP                  ;RETURN IF CERR W/O DATA LATE SET
      ERROR   4 ;OR 5, 6, 7 ;REPORT ALL OTHER ERRORS
      ERROR   13             ;REPORT DATA LATE
      ERROR   2              ;REPORT "OPERATION TO SLOW" OR "HEADER
                          ;0 NOT FOUND" MESSAGE
    
```



```

6400
6401 020216 104415          SCOPI          ;LOCAL LOOP TO 3$ ON ERROR
6402 020220 004437 040776  JSR          R4,GENCOM  ;GENERATE & COMPARE HEADERS
6403 020224 101200          101200        ;INCLUDING BAD SECTOR LISTS
6404 020226 000413          BR           4$        ;NO MISCOMPARES-SKIP
6405 020230 104015          ERROR        15        ;REPORT FIRST MISCOMPARE
6406
6407 020232 013700 001634  MOV          ERRMT,RO   ;GET ERROR LIMIT
6408 020236 005300          64$: DEC         RO    ;DECREMENT COUNT
6409 020240 001406          BEQ         65$        ;IF ZERO - EXIT
6410 020242 004437 040776  JSR          R4,GENCOM  ;CONTINUE DATA COMPARE
6411 020246 040000          40000
6412 020250 000402          BR           65$        ;NO MORE ERRORS - EXIT
6413 020252 104016          ERROR        16        ;REPORT NEXT ERROR
6414 020254 000770          BR           64$        ;LOOP
6415 020256
6416
6417 020256 104415          4$:  SCOPI          ;LOCAL LOOP TO 3$
6418 020260 005701          TST         R1         ;POINTING TO TRACK 1
6419 020262 001002          BNE         5$         ;NO-EXIT
6420 020264 005201          INC         R1         ;BUMP TO TRACK 1
6421 020266 000730          BR          3$         ;LOOP
6422
6423 020270
6424
6425
6426
6427
6428
6429
6430
6431
6432 020270 000004          5$: *****
6433 020272 012737 000062 001262  ;:*****
6434 020300 012737 000312 001664  ;:TEST 62      24 SECTOR FORMAT DATA TRANSFER (PART 1)
6435 020306 104416          ;:
6436 020310 104003          ;:
6437
6438 020312 004437 040776  ;:
6439 020316 000013          ;:  ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
6440 020320 000400          ;:  TO CYLINDER 312, TRACK 0, SECTOR 0.  READ SECTOR BACK
6441
6442 020322 004437 040776  ;:  AND MAKE SURE IT IS CORRECT.
6443 020326 002007          ;:*****
6444 020330 000400          ;:
6445
6446 020332 004437 034574  ;:
6447 020336 010123          ;:  SCOPE
6448 020340 177400          ;:  MOV      #50.,$TIMES  ;:DO 50. ITERATIONS
6449 020342 062604          ;:  MOV      #312,REFMT  ;:SET REFORMAT SWITCH
6450 020344 000          ;:  TSSINIT  ;:CLEAR SUBSYSTEM
6451 020345 000          ;:  ERROR    3  ;:BAD INIT ERROR
6452 020346 000312          ;:
6453
6454 020350 104417          ;:  JSR      R4,GENCOM  ;:GENERATE DATA
6455 020352 104430          ;:  13        ;:PATTERN 13
6456
6457
6458
6459
6460
6461
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476
6477
6478
6479
6480
6481
6482
6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555

```


E10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 121
T62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

SEQ 0121

```

6456 020354 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6457
6458 020356 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6459 020360 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6460
6461 020362 004437 034574    JSR    R4,LRLOAD ;LOAD "L" REGS
6462 020366 010121          RDDATA!CFMT     ;RDDATA!CFMT
6463 020370 177400          -400           ;-400 WORDS
6464 020372 060604          Ibuff          ;IBUFF IS BUFF ADDRESS
6465 020374 000           .BYTE 0         ;SECTOR 0
6466 020375 000           .BYTE 0         ;TRACK 0
6467 020376 000312          312           ;CYLINDER 312
6468
6469 020400 104417          TLOADRK        ;LOAD RK REGS
6470 020402 104424          TWAT32        ;WAIT FOR INTERRUPT
6471 020404 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
6472
6473 020406 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
6474 020410 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6475
6476 020412 004437 040776    JSR    R4,GENCOM ;COMPARE DATA
6477 020416 100000          100000
6478 020420 000400          400           ;400 WORDS
6479 020422 000413          BR    15       ;NO MISCOMPARES-SKIP
6480 020424 104015          ERROR 15       ;REPORT 1ST ERROR
6481
6482 020426 013700 001634    MOV    ERRLMT,RO ;GET ERROR LIMIT
6483 020432 005300          64$: DEC    RO   ;DECREMENT COUNT
6484 020434 001406          BEQ    65$     ;IF ZERO - EXIT
6485 020436 004437 040776    JSR    R4,GENCOM ;CONTINUE DATA COMPARE
6486 020442 040000          40000
6487 020444 000402          BR    65$     ;NO MORE ERRORS - EXIT
6488 020446 104016          ERROR 16       ;REPORT NEXT ERROR
6489 020450 000770          BR    64$     ;LOOP
6490 020452
6491
6492 020452
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506 020452 000004          1$: *****
6507 020454 012737 000062 001262  TST63: SCOPE *****
6508 020462 012737 000312 001664    MOV    #50, $TIMES ;:DO 50. ITERATIONS
6509 020470 032737 000200 001656    MOV    #312,REFMT ;SET REFORMAT SWITCH
6510 020476 001504          BIT    #PARBKO,OPTFLG ;PARITY OPTION PRESENT?
6511                                BEQ    4$       ;YES-SKIP

```

```

1$:
*****
*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 110 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.
*****

```


F10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 122
T63 24 SECTOR FORMAT DATA TRANSFER (PART 2)

SEQ 0122

```

6512 020500 004437 040776 1$: JSR R4,GENCOM ;GENERATE DATA
6513 020504 000014 14 ;PATTERN 14
6514 020506 000400 400 ;400 WORDS
6515
6516 020510 012746 000340 MOV #PR7,-(SP) ;SET PRIORITY TO 7
6517 020514 012746 020522 MOV #10$,-(SP) ;SET ADDRESS
6518 020520 000002 RTI
6519 020522 10$:
6520
6521 020522 032737 000100 001656 BIT #PARBK1,OPTFLG ;TEST IF BANK 1 AVAIL
6522 020530 001403 5$ ;NO - SKIP
6523 020532 012777 000004 161144 MOV #BIT2,@MMCSR2 ;SET WRONG PARITY WRITE
6524 020540 012777 000004 161134 5$: MOV #BIT2,@MMCSR1 ;SET WRONG PARITY BIT
6525 020546 012737 177777 063026 MOV #-1,OBUFF+222 ;WRITE WITH BAD PARITY
6526 020554 012737 177777 063024 MOV #-1,OBUFF+220
6527 020562 012777 000001 161112 MOV #BIT0,@MMCSR1 ;CLEAR WRONG PARITY, SET IE
6528 020570 032737 000100 001656 BIT #PARBK1,OPTFLG ;TEST IF BANK 1 AVAIL
6529 020576 001403 5$ ;NO - SKIP
6530 020600 012777 000001 161076 MOV #BIT0,@MMCSR2
6531
6532 020606 013746 001622 6$: MOV RKPRI,-(SP) ;RESTORE PRIORITY
6533 020612 012746 020620 MOV #11$,-(SP)
6534 020616 000002 RTI
6535 020620 11$:
6536 020620 104416 TSSINIT ;CLEAR SUBSYSTEM
6537 020622 104003 ERROR 3 ;BAD INIT ERROR
6538 020624 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
6539 020630 010123 WRDATA!CFMT ;WRDATA!CFMT
6540 020632 177400 -400 ;-400 WORDS
6541 020634 062604 OBUFF ;OBUFF IS BUFF ADDRESS
6542 020636 000 .BYTE 0 ;SECTOR 0
6543 020637 000 .BYTE 0 ;TRACK 0
6544 020640 000312 312 ;CYLINDER 312
6545
6546 020642 104417 TLOADRK ;LOAD RK REGS
6547 020644 104430 TWAT96 ;WAIT FOR INTERRUPT
6548 020646 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
6549
6550 020650 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
6551 020652 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6552 : A UNIBUS PARITY ERROR REPORTED AT THIS
6553 : TIME INDICATES CONTROLLER IS NOT
6554 : PROPERLY TRANSFERRING DATA IN 18 BIT MODE
6555
6556 020654 005077 161022 CLR @MMCSR1 ;CLEAR IE
6557 020660 032737 000100 001656 BIT #PARBK1,OPTFLG
6558 020666 001402 BEQ 7$
6559 020670 005077 161010 CLR @MMCSR2
6560 020674 005037 063026 7$: CLR OBUFF+222
6561 020700 005037 063024 CLR OBUFF+220
6562 020704 004737 033704 JSR PC,OPTTST ;RESET OPTIONS
6563
6564 020710 4$:
6565 :*****
6566 :*TEST 64 24 SECTOR FORMAT DATA TRANSFER (PART 3)
6567 :*

```


G10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 123
T64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

SEQ 0123

```

6568      : *      ISSUE A WRITE DATA OF 1000 WORDS IN 24 SECTOR FORMAT
6569      : *      TO CYLINDER 312, TRACK 0, SECTOR 23, READ SECTOR BACK
6570      : *      AND MAKE SURE IT IS CORRECT. MAKE SURE THAT MID-TRANSFER
6571      : *      SEEK HAS TAKEN PLACE.
6572      : *
6573      : *****
6574 020710 000004      TST64: SCOPE
6575 020712 012737 000062 001262      MOV      #50.,$TIMES      ;;DO 50. ITERATIONS
6576 020720 012737 000312 001664      MOV      #312,REFMT      ;;SET REFORMAT SWITCH
6577 020726 004737 033704      JSR      PC,OPTTST      ;;SET UP OPTIONS
6578 020732 104416      TSSINIT      ;;CLEAR SUBSYSTEM
6579 020734 104003      ERROR      3      ;;BAD INIT ERROR
6580
6581 020736 004437 040776      JSR      R4,GENCOM      ;GENERATE DATA
6582 020742 000015      15      ;PATTERN 15
6583 020744 001000      1000      ;1000 WORDS
6584
6585 020746 004437 040776      JSR      R4,GENCOM      ;CLEAR IBUFF TO ALL ONES
6586 020752 002007
6587 020754 001000      2007
6588      1000
6589 020756 004437 034574      JSR      R4,LRLoad      ;LOAD "L" REGS
6590 020762 010123      WRDATA!CFMT      ;WRDATA!CFMT
6591 020764 177000      -1000      ;-1000 WORDS
6592 020766 062604      OBUFF      ;OBUFF IS BUFF ADDRESS
6593 020770      023      .BYTE      23      ;SECTOR 23
6594 020771      000      .BYTE      0      ;TRACK 0
6595 020772 000312      312      ;CYLINDER 312
6596
6597 020774 104417      TLOADRK      ;LOAD RK REGS
6598 020776 104430      TWAT96      ;WAIT FOR INTERRUPT
6599 021000 104002      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
6600
6601 021002 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
6602 021004 104004      ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6603
6604 021006 004437 034574      JSR      R4,LRLoad      ;LOAD "L" REGS
6605 021012 010121      RDDATA!CFMT      ;RDDATA!CFMT
6606 021014 177000      -1000      ;-1000 WORDS
6607 021016 060604      IBUFF      ;IBUFF IS BUFF ADDRESS
6608 021020      023      .BYTE      23      ;SECTOR 23
6609 021021      000      .BYTE      0      ;TRACK 0
6610 021022 000312      312      ;CYLINDER 312
6611
6612 021024 104417      TLOADRK      ;LOAD RK REGS
6613 021026 104426      TWAT64      ;WAIT FOR INTERRUPT
6614 021030 104002      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
6615
6616 021032 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
6617 021034 104004      ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6618
6619 021036 004437 040776      JSR      R4,GENCOM      ;COMPARE DATA
6620 021042 100000      100000
6621 021044 001000      1000      ;1000 WORDS
6622 021046 000413      BR      1$      ;NO MISCOMPARES-SKIP
6623 021050 104015      ERROR      15      ;REPORT FIRST ERROR

```


H10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 124
T64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

SEG 0124

6624
6625 021052 013700 001634
6626 021056 005300
6627 021060 001406
6628 021062 004437 040776
6629 021066 040000
6630 021070 000402
6631 021072 104016
6632 021074 000770
6633 021076
6634
6635 021076
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649
6650
6651
6652

64\$: MOV ERRLMT,RO ;GET ERROR LIMIT
DEC RO ;DECREMENT COUNT
BEQ 65\$;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 65\$;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 64\$;LOOP

65\$:

1\$:

.SBTTL **SPECIAL DATA TRANSFER TESTS

```
*****  
*TEST 65 MULTI-SECTOR DATA TRANSFER AND BSE  
*  
* 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.  
*  
*****
```

6653 021076 000004
6654 021100 012737 000062 001262
6655 021106 012737 000312 001664
6656 021114 104416
6657 021116 104003
6658
6659 021120 004437 034574
6660 021124 000127
6661 021126 177676
6662 021130 062604
6663 021132 000
6664 021133 000
6665 021134 000312
6666
6667 021136 004437 040776
6668 021142 000600
6669
6670 021144 042737 040000 062614
6671 021152 042737 040000 062616
6672
6673 021160 104417
6674 021162 104431
6675 021164 104002
6676
6677 021166 104421
6678 021170 104004
6679

```
*****  
*TST65: SCOPE  
MOV #50,$TIMES ;DO 50. ITERATIONS  
MOV #312,REFMT ;SET REFORMAT SWITCH  
TSSINIT ;CLEAR SUBSYSTEM  
ERROR 3 ;BAD INIT ERROR  
*  
JSR R4,LRLoad ;LOAD "L" REGS  
WRHEAD ;WRHEAD  
-102 ;-102 WORDS  
OBUFF ;OBUFF IS BUFF ADDRESS  
.BYTE 0 ;SECTOR 0  
.BYTE 0 ;TRACK 0  
312 ;CYLINDER 312  
*  
JSR R4,GENCOM ;BUILD HEADERS  
600  
*  
BIC #BIT14,OBUFF+10 ;MARK SECTOR 1 BAD  
BIC #BIT14,OBUFF+12 ;CORRECT HURC  
*  
TLOADRK ;LOAD RK REGS  
TWAT112 ;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
```


6680	021172	004437	040776			JSR	R4,GENCOM	:GENERATE DATA
6681	021176	000016				16		:PATTERN 16
6682	021200	001000				1000		:1000 WORDS
6683								
6684	021202	004437	040776			JSR	R4,GENCOM	:CLEAR Ibuff TO ALL ONES
6685	021206	002007				2007		
6686	021210	001000				1000		
6687								
6688	021212	004437	034574			JSR	R4,LRLOAD	:LOAD "L" REGS
6689	021216	000123				WRDATA		:WRDATA
6690	021220	177000				-1000		: -1000 WORDS
6691	021222	062604				OBUFF		:OBUFF IS BUFF ADDRESS
6692	021224	000				.BYTE	0	:SECTOR 0
6693	021225	000				.BYTE	0	:TRACK 0
6694	021226	000312				312		:CYLINDER 312
6695								
6696	021230	104417				TLOADRK		:LOAD RK REGS
6697	021232	104424				TWAT32		:WAIT FOR INTERRUPT
6698	021234	104002				ERROR	2	:TO SLOW/NOT COMPLETE ERROR
6699								
6700	021236	104422				TCHKWE		:CHECK OPERATION WITH EXPECTED ERR
6701	021240	000000				0		
6702	021242	000100				BSERR		:BAD SECTOR ERROR
6703	021244	000000				0		
6704	021246	104004				ERROR	4: OR 5,6,7	:REPORT ALL DISCREPANCIES
6705	021250	005037	046520			CLR	GRP4ER	:CLEAR GROUP 4 ERRORS
6706	021254	004437	036366			JSR	R4,CHKCTS	:CHECK CYL, TRK, SECT CORRECT AFTER ABORTED WRITE
6707	021260	032737	000020	046520		BIT	#TRKERR,GRP4ER	:TRK IN ERROR?
6708	021266	001416				BEG	1\$:NO-SKIP
6709	021270	012737	053071	001450		MOV	#EM13,EM10N	: "TRACK ADDRESS INCORRECT"
6710	021276	013737	046474	001202		MOV	EXPTRK,\$REG10	:EXPECTED VALUE
6711	021304	013737	046506	001204		MOV	REALTRK,\$REG11	:REAL VALUE
6712	021312	012737	047205	057662		MOV	#OPER37,DF010A	: "AFTER WRITE DATA TERMINATED WITH BSE"
6713	021320	104010				ERROR	10	
6714	021322	000527				BR	5\$:EXIT
6715								
6716	021324	032737	000040	046520	1\$:	BIT	#SECERR,GRP4ER	:SECTOR IN ERROR?
6717	021332	001422				BEG	3\$:NO-SKIP
6718	021334	012737	053121	001450		MOV	#EM14,EM10N	: "SECTOR ADDRESS INCORRECT"
6719	021342	012737	047205	057662		MOV	#OPER37,DF010A	: "AFTER WRITE DATA ABORTED WITH BSE"
6720	021350	013737	046472	001202		MOV	EXPSEC,\$REG10	:EXPECTED VALUE
6721	021356	013737	046510	001204		MOV	REALSEC,\$REG11	:REAL VALUE
6722	021364	104010				ERROR	10	
6723	021366	000505				BR	5\$:EXIT
6724	021370	104415				SCOPI		:LOCAL LOOP TO BEGINNING OF TEST
6725	021372	012737	021400	001110		MOV	#3\$, \$LPERR	:SET LOCAL LOOP ON ERROR
6726	021400				3\$:			
6727	021400	104416				TSSINIT		:CLEAR SUBSYSTEM
6728	021402	104003				ERROR	3	:BAD INIT ERROR
6729	021404	004437	034574			JSR	R4,LRLOAD	:LOAD "L" REGS
6730	021410	000121				RDDATA		:RDDATA
6731	021412	177400				-400		: -400 WORDS
6732	021414	060604				IBUFF		:IBUFF IS BUFF ADDRESS
6733	021416	000				.BYTE	0	:SECTOR 0
6734	021417	000				.BYTE	0	:TRACK 0
6735	021420	000312				312		:CYLINDER 312

6736								
6737	021422	104417			TLOADRK			;LOAD RK REGS
6738	021424	104424			TWAT32			;WAIT FOR INTERRUPT
6739	021426	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
6740								
6741	021430	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
6742	021432	104004			ERROR	4 ;OR 5, 6, 7, 10		;REPORT ALL ERRORS
6743								
6744	021434	004437	040776		JSR	R4,GENCOM		;COMPARE DATA
6745	021440	100000			100000			
6746	021442	000400			400			;400 WORDS
6747	021444	000413			BR	4\$;NO MISCOMPARES-EXIT
6748	021446	104015			ERROR	15		;REPORT FIRST ERROR
6749								
6750	021450	013700	001634		MOV	ERRLMT,RO		;GET ERROR LIMIT
6751	021454	005300		64\$:	DEC	RO		;DECREMENT COUNT
6752	021456	001406			BEQ	65\$;IF ZERO - EXIT
6753	021460	004437	040776		JSR	R4,GENCOM		;CONTINUE DATA COMPARE
6754	021464	040000			40000			
6755	021466	000402			BR	65\$;NO MORE ERRORS - EXIT
6756	021470	104016			ERROR	16		;REPORT NEXT ERROR
6757	021472	000770			BR	64\$;LOOP
6758	021474			65\$:				
6759								
6760	021474	004437	040776	4\$:	JSR	R4,GENCOM		;CLEAR IBUFF
6761	021500	002007			2007			
6762	021502	001000			1000			
6763								
6764	021504	004437	034574		JSR	R4,LRLOAD		;LOAD "L" REGS
6765	021510	000121			RDDATA			;RDDATA
6766	021512	177000			-1000			; -1000 WORDS
6767	021514	060604			IBUFF			;IBUFF IS BUFF ADDRESS
6768	021516	000			.BYTE	0		;SECTOR 0
6769	021517	000			.BYTE	0		;TRACK 0
6770	021520	000312			312			;CYLINDER 312
6771								
6772	021522	104417			TLOADRK			;LOAD RK REGS
6773	021524	104424			TWAT32			;WAIT FOR INTERRUPT
6774	021526	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
6775								
6776	021530	104422			TCHKWE			;CHECK OPERATION WITH EXPECTED ERROR
6777	021532	000000			0			
6778	021534	000100			BSERR			;BAD SECTOR ERROR
6779	021536	000000			0			
6780	021540	104004			ERROR	4; OR 5,6,7		;REPORT ALL DISCREPANCIES
6781								
6782	021542	004437	040776		JSR	R4,GENCOM		;COMPARE DATA AGAIN
6783	021546	100000			100000			
6784	021550	000400			400			;400 WORDS
6785	021552	000413			BR	5\$;NO MISCOMPARES
6786	021554	104015			ERROR	15		;REPORT FIRST ERROR
6787								
6788	021556	013700	001634		MOV	ERRLMT,RO		;GET ERROR LIMIT
6789	021562	005300		66\$:	DEC	RO		;DECREMENT COUNT
6790	021564	001406			BEQ	67\$;IF ZERO - EXIT
6791	021566	004437	040776		JSR	R4,GENCOM		;CONTINUE DATA COMPARE


```

6848 021744 177400          -400          ; -400 WORDS
6849 021746 060604          Ibuff         ; Ibuff IS BUFF ADDRESS
6850 021750          000          .BYTE 0       ; SECTOR 0
6851 021751          000          .BYTE 0       ; TRACK 0
6852 021752 000312          312          ; CYLINDER 312
6853
6854 021754 110037 001606    MOVb         R0,L,DS ; LOAD SECTOR COUNTER INTO DESIRED SECTOR
6855
6856 021760 104417          TLOADRK      ; LOAD RK REGS
6857 021762 104424          TWAT32      ; WAIT FOR INTERRUPT
6858 021764 104002          ERROR 2     ; TO SLOW/NOT COMPLETE ERROR
6859
6860 021766 104421          TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
6861 021770 104004          ERROR 4 ;OR 5, 6, 7, 10 ; REPORT ALL ERRORS
6862
6863 021772 104415          SCOPI       ; LOCAL LOOP TO 3$ ON ERROR
6864
6865 021774 022700 000024    CMP         #24,R0 ; LAST SECTOR READ?
6866 022000 001402          BEQ         4$   ; YES-EXIT
6867 022002 005200          INC         R0   ; BUMP SECTOR COUNTER
6868 022004 000754          BR         3$   ; LOOP
6869
6870 022006 005037 001664    4$: CLR      REfmt ; CLEAR REFORMAT SWITCH
6871
6872          .SBTTL **WRITE CHECK TESTS
6873
6874          ;*****
6875          ;*TEST 67      WRITE-CHECK WITH NO ERROR
6876          ;*
6877          ;*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH A KNOWN PATTERN.
6878          ;*      DO A WRITE-CHECK OF 400 WORDS. MAKE SURE NO
6879          ;*      ERROR OCCURS.
6880          ;*
6881          ;*****
6882          †ST67: SCOPE
6883 022012 000004          MOV         #50.,$TIMES ; DO 50. ITERATIONS
6884 022014 012737 000062 001262 TSSINIT     ; CLEAR SUBSYSTEM
6885 022022 104416          ERROR 3     ; BAD INIT ERROR
6886
6887 022026 004437 034574    JSR         R4,LRLOAD ; LOAD "L" REGS
6888 022032 000123          WRDATA     ; WRDATA
6889 022034 177400          -400      ; -400 WORDS
6890 022036 062604          Obuff     ; Obuff IS BUFF ADDRESS
6891 022040          000      ; SECTOR 0
6892 022041          000      ; TRACK 0
6893 022042 000312          312      ; CYLINDER 312
6894
6895 022044 004437 040776    JSR         R4,GENCOM ; GENERATE DATA
6896 022050 000002          2         ; PATTERN 2
6897 022052 000400          400      ; 400 WORDS
6898
6899 022054 104417          TLOADRK   ; LOAD RK REGS
6900 022056 104430          TWAT96   ; WAIT FOR INTERRUPT
6901 022060 104002          ERROR 2   ; TO SLOW/NOT COMPLETE ERROR
6902
6903 022062 104421          TCHKOP   ; CHECK OPERATION FOR ANY ERRORS
    
```


M10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 129
T67 WRITE-CHECK WITH NO ERROR

SEQ 0129

```

6904 022064 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6905
6906 022066 004437 034574  JSR    R4,LRLOAD      ;LOAD "L" REGS
6907 022072 000131          WRTCHK          ;WRTCHK
6908 022074 177400          -400           ; -400 WORDS
6909 022076 062604          OBUFF          ;OBUFF IS BUFF ADDRESS
6910 022100 000           .BYTE 0        ;SECTOR 0
6911 022101 000           .BYTE 0        ;TRACK 0
6912 022102 000312          312           ;CYLINDER 312
6913
6914 022104 104417          TLOADRK        ;LOAD RK REGS
6915 022106 104424          TWAT32        ;WAIT FOR INTERRUPT
6916 022110 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
6917
6918 022112 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
6919 022114 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6920
6921 *****
6922 *TEST 70      WRITE CHECK ERROR (PART 1)
6923
6924 *          WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
6925 *          WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME
6926 *          DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
6927 *          CONFIGURATIONS:
6928 *
6929 *          000001 000020 000400 010000
6930 *          000002 000040 001000 020000
6931 *          000004 000100 002000 040000
6932 *          000010 000200 004000 100000
6933 *
6934 *          MAKE SURE WRITE CHECK ERROR SET FOR EACH
6935 *          OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
6936 *          AND WORD COUNT IS CORRECT.
6937 *
6938 *****
6939 †ST70:  SCOPE
6940      MOV    #50,$TIMES      ;DO 50. ITERATIONS
6941      MOV    #BIT0,R0        ;SET LO ORDER BIT IN R0 FOR
6942      TSSINIT                ;CAUSING WRITE CHECK ERROR
6943      ERROR 3                ;CLEAR SUBSYSTEM
6944      JSR    R4,GENCOM       ;BAD INIT ERROR
6945      1                      ;GENERATE DATA, ALL 0'S
6946      400
6947
6948 022146 004437 034574  JSR    R4,LRLOAD      ;LOAD "L" REGS
6949 022152 000123          WRDATA        ;WRDATA
6950 022154 177400          -400         ; -400 WORDS
6951 022156 062604          OBUFF        ;OBUFF IS BUFF ADDRESS
6952 022160 000           .BYTE 0        ;SECTOR 0
6953 022161 000           .BYTE 0        ;TRACK 0
6954 022162 000312          312         ;CYLINDER 312
6955
6956 022164 104417          TLOADRK        ;LOAD RK REGS
6957 022166 104430          TWAT96        ;WAIT FOR INTERRUPT
6958 022170 104002          ERROR 2       ;TO SLOW/NOT COMPLETE ERROR
6959

```


B11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRSKC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 131
T70 WRITE CHECK ERROR (PART 1)

SEQ 0131

```

7016 022334 013737 001544 001204      MOV    T.BA,$REG11      ;BAD VALUE
7017 022342 012737 047241 057662      MOV    #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7018 022350 104010                                ERROR  10
7019
7020 022352 023727 001542 177511 2$:    CMP    T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7021 022360 001460                                BEQ    3$              ;YES-SKIP
7022 022362 012737 052707 001450      MOV    #EM10,EM10N     ;"INCORRECT WC"
7023 022370 012737 047241 057662      MOV    #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7024 022376 012737 177511 001202      MOV    #-267,$REG10    ;GOOD VALUE
7025 022404 013737 001542 001204      MOV    T.WC,$REG11    ;ERROR VALUE
7026 022412 104010                                ERROR  10
7027 022414 000442                                BR     3$              ;EXIT
7028
7029 022416 023727 001544 063030 6$:    CMP    T.BA,#OBUFF+224 ;TEST IF BA AT HI SIDE
7030 022424 001415                                BEQ    7$              ;YES - SKIP
7031 022426 012737 052734 001450      MOV    #EM11,EM10N     ;SET MESSAGE
7032 022434 012737 063030 001202      MOV    #OBUFF+224,$REG10 ;GOOD VALUE
7033 022442 013737 001544 001204      MOV    T.BA,$REG11    ;ERROR VALUE
7034 022450 012737 047241 057662      MOV    #OPER41,DF010A  ;"WRITE CHECK ABORTED WITH WCE"
7035 022456 104010                                ERROR  10
7036
7037 022460 023727 001542 177512 7$:    CMP    T.WC,#-266      ;TEST IF WORD COUNT AT HI SIDE
7038 022466 001415                                BEQ    3$              ;YES - SKIP
7039 022470 012737 052707 001450      MOV    #EM10,EM10N     ;SET MESSAGE
7040 022476 012737 047241 057662      MOV    #OPER41,DF010A  ;"WC ABORTED WITH WCE"
7041 022504 012737 177512 001202      MOV    #-266,$REG10    ;GOOD VALUE
7042 022512 013737 001542 001204      MOV    T.WC,$REG11    ;ERROR VALUE
7043 022520 104010                                ERROR  10
7044
7045 022522 104415                                3$:    SCOPI              ;LOCAL LOOP ON ERROR TO 1$
7046
7047 022524 032700 100000                                BIT    #BIT15,RO       ;BIT 15 SET?
7048 022530 001002                                BNE   4$              ;YES-EXIT
7049 022532 006300                                ASL   RO              ;SHIFT ERROR BIT
7050 022534 000640                                BR    1$              ;LOOP
7051
7052 022536                                4$:
7053 :*****
7054 :*TEST 71      WRITE CHECK ERROR (PART 2)
7055 :*
7056 :*      WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
7057 :*      IN ALL WORDS.  WRITE CHECK CYLINDER 312, TRACK 0,
7058 :*      SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
7059 :*      ONE OF THE FOLLWOING CONFIGURATIONS:
7060 :*
7061 :*      177776 177757 177377 167777
7062 :*      177775 177737 176777 157777
7063 :*      177773 177677 175777 137777
7064 :*      177767 177577 173777 077777
7065 :*
7066 :*      MAKE SURE WRITE CHECK ERROR SET FOR EACH
7067 :*      OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
7068 :*      AND WORD COUNT IS CORRECT.
7069 :*
7070 :*****
7071 022536 000004      TST71:  SCOPE

```


7072	022540	012737	000062	001262	MOV	#50,\$TIMES	::DO 50. ITERATIONS
7073	022546	012700	177776		MOV	#177776,R0	::LOAD R0 FOR CAUSING WRITE CHECK ERROR
7074							
7075	022552	104416			TSSINIT		::CLEAR SUBSYSTEM
7076	022554	104003			ERROR	3	::BAD INIT ERROR
7077	022556	004437	040776		JSR	R4,GENCOM	::GENERATE DATA
7078	022562	000007			7		::ALL 1'S
7079	022564	000400			400		::400 WORDS
7080							
7081	022566	004437	034574		JSR	R4,LRLOAD	::LOAD "L" REGS
7082	022572	000123			WRDATA		::WRDATA
7083	022574	177400			-400		::-400 WORDS
7084	022576	062604			OBUFF		::OBUFF IS BUFF ADDRESS
7085	022600	000			.BYTE	0	::SECTOR 0
7086	022601	000			.BYTE	0	::TRACK 0
7087	022602	000312			312		::CYLINDER 312
7088							
7089	022604	104417			TLOADRK		::LOAD RK REGS
7090	022606	104430			TWAT96		::WAIT FOR INTERRUPT
7091	022610	104002			ERROR	2	::TO SLOW/NOT COMPLETE ERROR
7092							
7093	022612	104421			TCHKOP		::CHECK OPERATION FOR ANY ERRORS
7094	022614	104004			ERROR	4 ;OR 5, 6, 7, 10	::REPORT ALL ERRORS
7095							
7096	022616	004437	034574		JSR	R4,LRLOAD	::LOAD "L" REGS
7097	022622	000131			WRTCHK		::WRTCHK
7098	022624	177400			-400		::-400 WORDS
7099	022626	062604			OBUFF		::OBUFF IS BUFF ADDRESS
7100	022630	000			.BYTE	0	::SECTOR 0
7101	022631	000			.BYTE	0	::TRACK 0
7102	022632	000312			312		::CYLINDER 312
7103							
7104	022634	104417			TLOADRK		::LOAD RK REGS
7105	022636	104424			TWAT32		::WAIT FOR INTERRUPT
7106	022640	104002			ERROR	2	::TO SLOW/NOT COMPLETE ERROR
7107							
7108	022642	104421			TCHKOP		::CHECK OPERATION FOR ANY ERRORS
7109	022644	104004			ERROR	4 ;OR 5, 6, 7, 10	::REPORT ALL ERRORS
7110							
7111	022646	104415			SCOPI		::LOCAL LOOP TO START OF TEST
7112	022650	012737	022656	001110	MOV	#15,\$LPERR	::SET LOCAL LOOP
7113							
7114	022656	010037	063024		MOV	R0,OBUFF+220	::PUT WORD IN OBUFF TO CAUSE WCE
7115	022662	104416		15:	TSSINIT		::CLEAR SUBSYSTEM
7116	022664	104003			ERROR	3	::BAD INIT ERROR
7117							
7118	022666	004437	034574		JSR	R4,LRLOAD	::LOAD "L" REGS
7119	022672	000131			WRTCHK		::WRTCHK
7120	022674	177400			-400		::-400 WORDS
7121	022676	062604			OBUFF		::OBUFF IS BUFF ADDRESS
7122	022700	000			.BYTE	0	::SECTOR 0
7123	022701	000			.BYTE	0	::TRACK 0
7124	022702	000312			312		::CYLINDER 312
7125							
7126	022704	104417			TLOADRK		::LOAD RK REGS
7127	022706	104424			TWAT32		::WAIT FOR INTERRUPT

D11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 133
T71 WRITE CHECK ERROR (PART 2)

SEQ 0133

7128	022710	104002				ERROR 2	;TO SLOW/NOT COMPLETE ERROR
7129							
7130	022712	104422				TCHKWE	;CHECK OPERATION WITH EXPECTED ERROR
7131	022714	000000				0	
7132	022716	000004				WCKERR	;WRITE CHECK ERROR
7133	022720	000000				0	
7134	022722	104004				ERROR 4; OR 5,6,7	;REPORT ALL DISCREPANCIES
7135							
7136	022724	104415				SCOPI	;LOCAL LOOP TO IS
7137							
7138							
7139							
7140							
7141							
7142							
7143							
7144							
7145							
7146	022726	023727	001544	063026		CMP T.BA,#0BUFF+222	;CHECK BA HALT AT PROPER PLACE
7147	022734	001416				BEQ 2\$;YES-SKIP
7148	022736	101037				BHI 6\$;IF TO HI - SKIP
7149	022740	012737	052734	001450		MOV #EM11,EM10N	;"INCORRECT BA"
7150	022746	012737	063026	001202		MOV #0BUFF+222,\$REG10	;GOOD VALUE
7151	022754	013737	001544	001204		MOV T.BA,\$REG11	;BAD VALUE
7152	022762	012737	047241	057662		MOV #OPER41,DF010A	;"WRITE CHECK ABORTED WITH WCE"
7153	022770	104010				ERROR 10	
7154							
7155	022772	023727	001542	177511	2\$:	CMP T.WC,#-267	;CHECK WORD COUNT AT CORRECT VALUE
7156	023000	001460				BEQ 3\$;YES-SKIP
7157	023002	012737	052707	001450		MOV #EM10,EM10N	;"INCORRECT WC"
7158	023010	012737	047241	057662		MOV #OPER41,DF010A	;"WRITE CHECK ABORTED WITH WCE"
7159	023016	012737	177511	001202		MOV #-267,\$REG10	;GOOD VALUE
7160	023024	013737	001542	001204		MOV T.WC,\$REG11	;ERROR VALUE
7161	023032	104010				ERROR 10	
7162	023034	000442				BR 3\$;EXIT
7163							
7164	023036	023727	001544	063030	6\$:	CMP T.BA,#0BUFF+224	;TEST IF BA AT HI SIDE
7165	023044	001415				BEQ 7\$;YES - SKIP
7166	023046	012737	052734	001450		MOV #EM11,EM10N	;SET MESSAGE
7167	023054	012737	063030	001202		MOV #0BUFF+224,\$REG10	;GOOD VALUE
7168	023062	013737	001544	001204		MOV T.BA,\$REG11	;ERROR VALUE
7169	023070	012737	047241	057662		MOV #OPER41,DF010A	;"WRITE CHECK ABORTED WITH WCE"
7170	023076	104010				ERROR 10	
7171							
7172	023100	023727	001542	177512	7\$:	CMP T.WC,#-266	;TEST IF WORD COUNT AT HI SIDE
7173	023106	001415				BEQ 3\$;YES - SKIP
7174	023110	012737	052707	001450		MOV #EM10,EM10N	;SET MESSAGE
7175	023116	012737	047241	057662		MOV #OPER41,DF010A	;"WC ABORTED WITH WCE"
7176	023124	012737	177512	001202		MOV #-266,\$REG10	;GOOD VALUE
7177	023132	013737	001542	001204		MOV T.WC,\$REG11	;ERROR VALUE
7178	023140	104010				ERROR 10	
7179							
7180	023142	104415			3\$:	SCOPI	
7181							
7182	023144	032700	100000			BIT #BIT15,RO	;BIT 15 SET? (ALL PATTERNS TESTED)
7183	023150	001002				BNE 4\$;YES-EXIT

NOTE: THE WORD COUNT AND BUS ADDRESS CAN BE EITHER OF TWO 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 ONE 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 REFLECT THE ACTUAL WORD WHERE THE ERROR WAS.

E11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 134
T71 WRITE CHECK ERROR (PART 2)

SEQ 0134

```

7184 023152 006300          ASL      R0          ;SHIFT FOR NEXT TEST
7185 023154 000640          BR       1$          ;LOOP
7186
7187 023156
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198 023156 000004          *
7199 023160 012737 000062 001262  *
7200 023166 104416          *
7201 023170 104003          *
7202
7203 023172 004437 040776          JSR      R4,GENCOM  ;GENERATE DATA
7204 023176 000007          7          ;ALL 1'S
7205 023200 000400          400        ;400 WORDS
7206
7207 023202 004437 034574          JSR      R4,LRLOAD  ;LOAD "L" REGS
7208 023206 000123          WRDATA    ;WRDATA
7209 023210 177400          -400      ;-400 WORDS
7210 023212 062604          OBUFF     ;OBUFF IS BUFF ADDRESS
7211 023214 000          .BYTE    0      ;SECTOR 0
7212 023215 000          .BYTE    0      ;TRACK 0
7213 023216 000312          312      ;CYLINDER 312
7214
7215 023220 104417          TLOADRK   ;LOAD RK REGS
7216 023222 104430          TWAT96    ;WAIT FOR INTERRUPT
7217 023224 104002          ERROR    2      ;TO SLOW/NOT COMPLETE ERROR
7218
7219 023226 104421          TCHKOP    ;CHECK OPERATION FOR ANY ERRORS
7220 023230 104004          ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7221
7222 023232 005037 063026          CLR      OBUFF+222
7223
7224 023236 004437 034574          JSR      R4,LRLOAD  ;LOAD "L" REGS
7225 023242 000131          WRTCHK    ;WRTCHK
7226 023244 177670          -110     ;-110 WORDS
7227 023246 062604          OBUFF     ;OBUFF IS BUFF ADDRESS
7228 023250 000          .BYTE    0      ;SECTOR 0
7229 023251 000          .BYTE    0      ;TRACK 0
7230 023252 000312          312      ;CYLINDER 312
7231
7232 023254 104417          TLOADRK   ;LOAD RK REGS
7233 023256 104424          TWAT32    ;WAIT FOR INTERRUPT
7234 023260 104002          ERROR    2      ;TO SLOW/NOT COMPLETE ERROR
7235
7236 023262 104421          TCHKOP    ;CHECK OPERATION FOR ANY ERRORS
7237 023264 104004          ERROR    4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7238
7239

```

.SBTTL **MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT

F11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 135
**MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT

SEQ 0135

7240
7241
7242
7243
7244
7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295

023266	000004		
023270	012737	000005	001262
023276	032737	000400	001656
023304	001043		
023306	105037	023331	
023312	005037	023332	
023316	004437	034574	
023322	000127		
023324	177676		
023326	062604		
023330	000		
023331	000		
023332	000000		
023334	004437	040776	
023340	001200		
023342	104417		
023344	104431		
023346	104002		
023350	104421		
023352	104004		
023354	123727	023331	000002
023362	001403		
023364	105237	023331	
023370	000752		
023372	105037	023331	
023376	023727	023332	000003
023404	001403		
023406	005237	023332	
023412	000741		

*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.

TEST73:	SCOPE		
	MOV	#5, \$TIMES	:: DO 5 ITERATIONS
	BIT	#FPFMT, OPTFLG	:: TEST IF FIRST PASS SWITCH SET
	BNE	24\$:: YES - SKIP FORMAT
	CLRB	21\$:: CLEAR ADDRESS POINTERS
	CLR	22\$	
20\$:	JSR	R4, LRLoad	:: LOAD "L" REGISTERS
	WRHEAD		:: WRITE HEADER
	-102		:: 102 WORDS
	OBUFF		:: OBUFF IS BUFF ADDRESS
	.BYTE	0	:: SECTOR 0
21\$:	.BYTE	0	:: TRACK ADDRESS (VARIABLE)
22\$:	0		:: CYLINDER 0 (VARIABLE)
	JSR	R4, GENCOM	:: GO GENERATE HEADERS
	1200		:: WITH BAD SECTOR ERRORS
	TLOADRK		:: LOAD RK REGS
	TWAT112		:: 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
	CMPB	21\$, #2	:: TEST IF LAST TRACK
	BEQ	23\$:: YES - SKIP
	INCB	21\$:: ELSE BUMP TRACK
	BR	20\$:: LOOP
23\$:	CLRB	21\$:: CLEAR TRACK POINTER
	CMP	22\$, #3	:: TEST IF LAST CYLINDER WRITTEN
	BEQ	24\$:: YES - SKIP
	INC	22\$:: ELSE BUMP CYLINDER
	BR	20\$:: LOOP

G11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 136
T73 MAXIMUM DATA TRANSFER (PART 1)

SEQ 0136

```

7296
7297 023414 013737 023426 001110 24$: MOV 1$, $LPERR ;SET LOCAL LOOP ON ERROR
7298 023422 012703 000400          MOV #400, R3 ;SET COUNT FOR SECTOR CLEARING
7299 023426          1$:
7300 023426 104416          TSSINIT ;CLEAR SUBSYSTEM
7301 023430 104003          ERROR 3 ;BAD INIT ERROR
7302 023432 004437 034574          JSR R4, LRLOAD ;LOAD "L" REGS
7303 023436 000123          WRDATA ;WRDATA
7304 023440 177400          -400 ; -400 WORDS
7305 023442 062604          OBUFF ;OBUFF IS BUFF ADDRESS
7306 023444          .BYTE 0 ;SECTOR 0
7307 023445          .BYTE 0 ;TRACK 0
7308 023446 000000          0 ;CYLINDER 0
7309
7310 023450 104417          TLOADRK ;LOAD RK REGS
7311 023452 104434          TWAT159 ;WAIT FOR INTERRUPT
7312 023454 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7313
7314 023456 104421          TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7315 023460 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7316
7317 023462 104415          SCOPI ;LOCAL LOOP ON ERROR TO 1$
7318 023464 005303          DEC R3 ;DECREMENT COUNT
7319 023466 012762 062604 000004 2$: MOV #OBUFF, RKBA(R2) ;SET BA
7320 023474 012762 177400 000002          MOV #-400, RKWC(R2) ;AND WC AGAIN
7321 023502 005037 001662          CLR INTSET ;CLEAR INTERRUPT FLAG
7322 023506 013762 001626 000010          MOV DRVNUM, RKCS2(R2) ;SET DRIVE NUMBER
7323 023514 012762 000123 000000          MOV #WRDATA, RKCS1(R2) ;DO WRITE DATA
7324
7325 023522 104425          TWAT48 ;WAIT FOR INTERRUPT
7326 023524 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7327
7328 023526 032762 000200 000014          BIT #BSE, RKER(R2) ;BAD SECTOR ERROR?
7329 023534 001415          BEQ 3$ ;NO-SKIP
7330 023536 032737 000200 001656          BIT #BSE, OPTFLG ;TEST IF BSE TO MANY HAS BEEN REPORTED
7331 023544 001007          BNE 5$ ;YES - SKIP
7332 023546 052737 000200 001656          BIS #BSE, OPTFLG ;SET FLAG
7333 023554 012737 051362 001360          MOV #OPRO17, EMIN ;SET MESSAGE
7334 023562 104001          ERROR 1 ;"FIRST 256 SECTOR NOT BSE FREE"
7335 023564 000137 024120          5$: JMP 14$ ;GO TO EXIT
7336          3$: CHECK FOR ANY OTHER ERRORS
7337
7338 023570 104421          TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7339 023572 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7340 023574 104415          SCOPI ;LOCAL LOOP TO 1$ (RESTART SECTOR CLEAR)
7341
7342 023576 005303          DEC R3 ;DECREMENT COUNT
7343 023600 001332          BNE 2$ ;LOOP IF NOT ZERO
7344
7345 023602 004437 034574          JSR R4, LRLOAD ;LOAD "L" REGS
7346 023606 000117          SEEK ;SEEK
7347 023610 000000          0 ;0 WORDS
7348 023612 000000          0 ;0 IS BUFF ADDRESS
7349 023614          .BYTE 0 ;SECTOR 0
7350 023615          .BYTE 0 ;TRACK 0
7351 023616 000000          0 ;CYLINDER 0
    
```



```

7408 023776 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
7409 024000 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7410                                SCOPI          ;LOCAL LOOP ON ERROR TO 8$
7411 024002 104415          JSR R4,GENCOM  ;COMPARE DATA
7412                                100000
7413 024004 004437 040776 400          ;400 WORDS
7414 024010 100000          BR 13$          ;NO MISCOMPARE-EXIT LOOP
7415 024012 000400          ERROR 15       ;REPORT FIRST ERROR
7416 024014 000413
7417 024016 104015
7418                                MOV ERLMT,RO    ;GET ERROR LIMIT
7419 024020 013700 001634 64$: DEC RO        ;DECREMENT COUNT
7420 024024 005300          BEQ 65$        ;IF ZERO - EXIT
7421 024026 001406          JSR R4,GENCOM  ;CONTINUE DATA COMPARE
7422 024030 004437 040776 40000
7423 024034 040000          BR 65$        ;NO MORE ERRORS - EXIT
7424 024036 000402          ERROR 16       ;REPORT NEXT ERROR
7425 024040 104016          BR 64$        ;LOOP
7426 024042 000770
7427 024044 65$:
7428                                SCOPI          ;LOCAL LOOP TO 8$
7429 024044 104415 13$:
7430                                DEC R3         ;DEC READ LOOP COUNT
7431 024046 005303          BEQ 14$        ;IF ZERO-EXIT
7432 024050 001423
7433                                INCB 10$       ;BUMP SECTOR
7434 024052 105237 023764  CMPB 10$,#26   ;FINISHED WITH TRACK?
7435 024056 123727 023764 000026 BNE 9$        ;NO-LOOP
7436 024064 001332          CLRB 10$      ;CLEAR SECTOR
7437 024066 105037 023764  INCB 11$       ;BUMP TRACK
7438 024072 105237 023765  CMPB 11$,#3    ;FINISHED WITH CYLINDER?
7439 024076 123727 023765 000003 BNE 9$        ;NO-LOOP
7440 024104 001322          CLRB 11$     ;CLEAR TRACK
7441 024106 105037 023765  INC 12$        ;BUMP CYL.
7442 024112 005237 023766  BR 9$          ;LOOP
7443 024116 000715
7444
7445 024120 14$:

```

```

*****
*TEST 74          MAXIMUM DATA TRANSFER (PART 2)
*
* 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.
*****
TST74:  SCOPE
MOV #5,$TIMES          ;;DO 5 ITERATIONS

```

```

7452 024120 000004
7453 024122 012737 000005 001262

```


K11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 140
T74 MAXIMUM DATA TRANSFER (PART 2)

SEQ 0140

7520	024316	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
7521						
7522	024320	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
7523	024324	000105		CLEAR		;CLEAR
7524	024326	000000		0		;0 WORDS
7525	024330	000000		0		;0 IS BUFF ADDRESS
7526	024332	000		.BYTE	0	;SECTOR 0
7527	024333	000		.BYTE	0	;TRACK 0
7528	024334	000000		0		;CYLINDER 0
7529						
7530	024336	104417		TLOADRK		;LOAD RK REGS
7531	024340	104423		TWAT16		;WAIT FOR INTERRUPT
7532	024342	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7533						
7534	024344	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7535	024346	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
7536						
7537	024350	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
7538	024354	000123		WRDATA		;WRDATA
7539	024356	000000		0		;0 WORDS
7540	024360	062604		OBUFF		;OBUFF IS BUFF ADDRESS
7541	024362	000		.BYTE	0	;SECTOR 0
7542	024363	000		.BYTE	0	;TRACK 0
7543	024364	000000		0		;CYLINDER 0
7544						
7545	024366	012737	135143 062604	MOV	#135143,OBUFF	;SET WORD FOR OUTPUT
7546	024374	012700	000621	MOV	#401,R0	;SET COUNT FOR INTERRUPT WAIT
7547	024400	052737	000020 001610	BIS	#BAI,L.CS2	;SET BUS ADDRESS INC INHIBIT
7548						
7549	024406	104417		TLOADRK		;LOAD RK REGS
7550	024410	104434	5\$:	TWAT159		;WAIT FOR INTERRUPT
7551	024412	000401		BR	6\$;NO INTERRUPT-BRANCH
7552	024414	000403		BR	7\$;INTERRUPT-BRANCH
7553						
7554	024416	005300	6\$:	DEC	R0	;DEC WAIT COUNT
7555	024420	001373		BNE	5\$;LOOP IF NOT ZERO
7556	024422	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7557						
7558	024424		7\$:			
7559	024424	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7560	024426	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7561	024430	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
7562	024434	000117		SEEK		;SEEK
7563	024436	000000		0		;0 WORDS
7564	024440	000000		0		;0 IS BUFF ADDRESS
7565	024442	000		.BYTE	0	;SECTOR 0
7566	024443	000		.BYTE	0	;TRACK 0
7567	024444	000632		632		;CYLINDER 632
7568	024446	104417		TLOADRK		;LOAD RK REGS
7569	024450	104423		TWAT16		;WAIT FOR INTERRUPT
7570	024452	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7571	024454	005037	001662	CLR	INTSET	;CLEAR INTERRUPT FLAG
7572						
7573	024460	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7574	024462	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
7575						

L11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 141
T74 MAXIMUM DATA TRANSFER (PART 2)

SEG 0141

```

7576 024464 104427          TWAT80          ;WAIT FOR SECOND INIT
7577 024466 104002          ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
7578 024470 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
7579 024472 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7580
7581 024474 004437 034574    JSR R4,LRLOAD    ;LOAD "L" REGS
7582 024500 000105          CLEAR          ;CLEAR
7583 024502 000000          0              ;0 WORDS
7584 024504 000000          0              ;0 IS BUFF ADDRESS
7585 024506 000          .BYTE 0         ;SECTOR 0
7586 024507 000          .BYTE 0         ;TRACK 0
7587 024510 000000          0              ;CYLINDER 0
7588
7589 024512 104417          TLOADRK        ;LOAD RK REGS
7590 024514 104423          TWAT16         ;WAIT FOR INTERRUPT
7591 024516 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7592
7593 024520 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
7594 024522 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7595
7596 024524 004437 034574    JSR R4,LRLOAD    ;LOAD "L" REGS
7597 024530 000131          WRTCHK        ;WRTCHK
7598 024532 000000          0              ;0 WORDS
7599 024534 062604          OBUFF         ;OBUFF IS BUFF ADDRESS
7600 024536 000          .BYTE 0         ;SECTOR 0
7601 024537 000          .BYTE 0         ;TRACK 0
7602 024540 000000          0              ;CYLINDER 0
7603 024542 052737 000020 001610  BIS #BAI,L.CS2  ;SET BAI FLAG
7604 024550 012700 000621    MOV #401.,RO    ;SET WAIT COUNT
7605
7606 024554 104417          TLOADRK        ;LOAD RK REGS
7607 024556 104434          TWAT159       ;WAIT FOR INTERRUPT
7608 024560 000401          BR 9$         ;NO INTERRUPT-SKIP
7609 024562 000403          BR 10$        ;INTERRUPT-SKIP
7610
7611 024564 005300          9$: DEC RO      ;DEC WAIT COUNT
7612 024566 001373          BNE 8$        ;NOT ZERO-LOOP
7613 024570 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7614
7615 024572          10$:
7616 024572 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
7617 024574 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7618
7619 024576          11$:
7620          ;*****
7621          ;*TEST 75 CONTROLLER TIME OUT
7622          ;*
7623          ;* SEEK TO CYLINDER 632. ISSUE A RECALIBRATE AND DO NOT
7624          ;* WAIT FOR SECOND INTERRUPT. NOW ISSUE A READ HEADER
7625          ;* OF CYLINDER 0, TRACK 0. MAKE SURE CONTROLLER TIME
7626          ;* OUT SETS.
7627          ;*
7628          ;*****
7629 024576 000004          TST75: SCOPE
7630 024600 012737 000005 001262  MOV #5.,$TIMES ;:DO 5. ITERATIONS
7631 024606 104416          TSSINIT        ;CLEAR SUBSYSTEM
    
```


7632	024610	104003		ERROR	3		;BAD INIT ERROR
7633							
7634	024612	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
7635	024616	000117		SEEK			;SEEK
7636	024620	000000		0			;0 WORDS
7637	024622	000000		0			;0 IS BUFF ADDRESS
7638	024624	000		.BYTE	0		;SECTOR 0
7639	024625	000		.BYTE	0		;TRACK 0
7640	024626	000632		632			;CYLINDER 632
7641							
7642	024630	104417		TLOADRK			;LOAD RK REGS
7643	024632	104423		TWAT16			;WAIT FOR INTERRUPT
7644	024634	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7645							
7646	024636	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7647	024640	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7648							
7649	024642	005037	001662	CLR	INTSET		;CLEAR INTERRUPT FLAG
7650	024646	104427		TWAT80			;WAIT FOR SECOND INTERRUPT
7651	024650	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7652	024652	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7653	024654	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7654							
7655	024656	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
7656	024662	000105		CLEAR			;CLEAR
7657	024664	000000		0			;0 WORDS
7658	024666	000000		0			;0 IS BUFF ADDRESS
7659	024670	000		.BYTE	0		;SECTOR 0
7660	024671	000		.BYTE	0		;TRACK 0
7661	024672	000000		0			;CYLINDER 0
7662							
7663	024674	104417		TLOADRK			;LOAD RK REGS
7664	024676	104423		TWAT16			;WAIT FOR INTERRUPT
7665	024700	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7666							
7667	024702	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7668	024704	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7669							
7670	024706	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
7671	024712	000113		RECAL			;RECAL
7672	024714	000000		0			;0 WORDS
7673	024716	000000		0			;0 IS BUFF ADDRESS
7674	024720	000		.BYTE	0		;SECTOR 0
7675	024721	000		.BYTE	0		;TRACK 0
7676	024722	000000		0			;CYLINDER 0
7677							
7678	024724	104417		TLOADRK			;LOAD RK REGS
7679	024726	104423		TWAT16			;WAIT FOR INTERRUPT
7680	024730	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7681							
7682	024732	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
7683	024736	000125		RDHEAD			;RDHEAD
7684	024740	000000		0			;0 WORDS
7685	024742	000000		0			;0 IS BUFF ADDRESS
7686	024744	000		.BYTE	0		;SECTOR 0
7687	024745	000		.BYTE	0		;TRACK 0


```

7688 024746 000000 0 ;CYLINDER 0
7689
7690 024750 104417 TLOADRK ;LOAD RK REGS
7691 024752 104436 TWAT2S ;WAIT FOR INTERRUPT
7692 024754 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7693
7694 024756 104422 TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR
7695 024760 000000 0
7696 024762 000000 0
7697 024764 000002 CTOERR ;CONTROLLER TIME OUT
7698 024766 104004 ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
7699 024770 104416 TSSINIT ;CLEAR SUBSYSTEM
7700 024772 104003 ERROR 3 ;BAD INIT ERROR
7701 024774 005037 CLR INTSET ;CLEAR INT FLAG
7702 025000 012762 MOV #IE,RKCS1(R2) ;SET INT ENABLE
7703 025006 104437 TWAT8S ;WAIT FOR SECOND INT
7704 025010 104002 ERROR 2
7705
7706
7707
7708
7709
7710
7711
7712
7713
7714
7715
7716
7717
7718
7719
7720 025012 000004
7721 025014 012737 000012 001262
7722
7723 025022 104416 TSSINIT ;CLEAR SUBSYSTEM
7724 025024 104003 ERROR 3 ;BAD INIT ERROR
7725
7726 025026 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
7727 025032 000113 RECAL ;RECAL
7728 025034 000000 0 ;0 WORDS
7729 025036 000000 0 ;0 IS BUFF ADDRESS
7730 025040 000 .BYTE 0 ;SECTOR 0
7731 025041 000 .BYTE 0 ;TRACK 0
7732 025042 000000 0 ;CYLINDER 0
7733
7734 025044 104417 TLOADRK ;LOAD RK REGS
7735 025046 104423 TWAT16 ;WAIT FOR INTERRUPT
7736 025050 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7737
7738 025052 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
7739 025056 104437 TWAT8S ;WAIT FOR SECOND INTERRUPT
7740 025060 104002 ERROR 2
7741
7742 025062 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
7743 025064 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS

```

.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
: CLEAR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 1,
: TRACK 0, HEAD 0. SEEK INCOMPLETE BECAUSE OF OUTER
: LIMIT SHOULD BE THE ONLY ERROR SET.
*****

```

```

TST76: SCOPE
MOV #10.,$TIMES ;;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REGS
RECAL ;RECAL
0 ;0 WORDS
0 ;0 IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0
TLOADRK ;LOAD RK REGS
TWAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
CLR INTSET ;CLEAR INTERRUPT FLAG
TWAT8S ;WAIT FOR SECOND INTERRUPT
ERROR 2
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS

```



```

7744
7745 025066 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
7746 025072 000117 SEEK ;SEEK
7747 025074 000000 0 ;0 WORDS
7748 025076 000000 0 ;0 IS BUFF ADDRESS
7749 025100 000 .BYTE 0 ;SECTOR 0
7750 025101 000 .BYTE 0 ;TRACK 0
7751 025102 000002 2 ;CYLINDER 2
7752 025104 012737 000020 001616 MOV #PAT,L.MR1 ;SET EVEN PARITY BIT
7753 025112 104417 TLOADRK ;LOAD RK REGS
7754 025114 104423 TWAT16 ;WAIT FOR INTERRUPT
7755 025116 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7756
7757 025120 104416 TSSINIT ;CLEAR SUBSYSTEM
7758 025122 104003 ERROR 3 ;BAD INIT ERROR
7759
7760 025124 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
7761 025130 000123 WRDATA ;WRDATA
7762 025132 177400 -400 ;-400 WORDS
7763 025134 062604 OBUFF ;OBUFF IS BUFF ADDRESS
7764 025136 000 .BYTE 0 ;SECTOR 0
7765 025137 000 .BYTE 0 ;TRACK 0
7766 025140 000001 1 ;CYLINDER 1
7767
7768 025142 104417 TLOADRK ;LOAD RK REGS
7769 025144 104423 TWAT16 ;WAIT FOR INTERRUPT
7770 025146 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
7771
7772 025150 104422 TCHKWE ;CHECK OPERATION WITH ERROR
7773 025152 000002 SKIERR ;SEEK INCOMPLETE
7774 025154 000000 0
7775 025156 000000 0
7776 025160 104004 ERROR 4 ;OR 5,6,OR7 ;REPORT ALL DISCREPANCIES
7777
7778 ; THIS DELAY IS NEEDED TO ALLOW SEEK AND NO MOTION TO
7779 ; SET IN THE DRIVE BEFORE THE SUBSYSTEM CLEAR IS DONE.
7780
7781 025162 012701 025000 MOV #25000,R1 ;SET A COUNT
7782 025166 005301 3$: DEC R1 ;DEC COUNT
7783 025170 001376 BNE 3$ ;LOOP UNTIL ZERO
7784
7785 025172 104416 TSSINIT ;CLEAR SUBSYSTEM
7786 025174 104003 ERROR 3 ;BAD INIT ERROR
7787
7788 025176 012762 000100 000000 MOV #IE,RKCS1(R2) ;SET IE
7789
7790 025204 005037 001662 CLR INTSET ;CLEAR INT FLAG
7791 025210 104437 TWAT8S ;WAIT FOR SECOND INTERRUPT
7792 025212 000401 BR 1$
7793 025214 000404 BR 2$
7794
7795 025216 012737 057175 001372 1$: MOV #DH016,DH2N ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
7796 ;ALLOWING HEADS TO RELOAD"
7797 025224 104002 ERROR 2
7798
7799 025226 2$:

```


7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824
7825
7826
7827
7828
7829
7830
7831
7832
7833
7834
7835
7836
7837
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855

025226 000004
025230 012737 000062 001262
025236 104416
025240 104003
025242 004437 034574
025246 000121
025250 177400
025252 060604
025254 000
025255 000
025256 000000
025260 104417
025262 012762 000001 000022
025270 104423
025272 104002
025274 104422
025276 000000
025300 000000
025302 000020
025304 104004
025306 000004
025310 012737 000062 001262
025316 104416
025320 104003
025322 004437 040776
025326 000007
025330 000400

```
*****  
*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.  
*****  
†ST77: SCOPE  
MOV #50.,$TIMES ;DO 50. ITERATIONS  
TSSINIT ;CLEAR SUBSYSTEM  
ERROR 3 ;BAD INIT ERROR  
  
JSR R4,LRLoad ;LOAD "L" REGS  
RDATA ;RDATA  
-400 ;-400 WORDS  
IBUFF ;IBUFF IS BUFF ADDRESS  
.BYTE 0 ;SECTOR 0  
.BYTE 0 ;TRACK 0  
0 ;CYLINDER 0  
  
TLOADRK ;LOAD RK REGS  
  
MOV #1,RKSPAR(R2) ;WRITE SPARE REGISTER  
  
TWAT16 ;WAIT FOR INTERRUPT  
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR  
  
TCHKWE ;CHECK OPERATION WITH EXPECTED ERROR  
0  
0  
PGERR ;PROG ERROR  
ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES  
  
*****  
*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.  
*****  
†ST100: SCOPE  
MOV #50.,$TIMES ;DO 50. ITERATIONS  
TSSINIT ;CLEAR SUBSYSTEM  
ERROR 3 ;BAD INIT ERROR  
  
JSR R4,GENCOM ;GENERATE DATA OF ALL ONES  
7  
400
```


7856	025332	004437	034574		JSR	R4,LRLOAD		;LOAD "L" REGS
7857	025336	000123			WRDATA			;WRDATA
7858	025340	177400			-400			; -400 WORDS
7859	025342	062604			OBUFF			;OBUFF IS BUFF ADDRESS
7860	025344	000			.BYTE	0		;SECTOR 0
7861	025345	000			.BYTE	0		;TRACK 0
7862	025346	000000			0			;CYLINDER 0
7863								
7864	025350	104417			TLOADRK			;LOAD RK REGS
7865	025352	104430			TWAT96			;WAIT FOR INTERRUPT
7866	025354	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7867								
7868	025356	104421			TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7869	025360	104004			ERROR	4 ;OR 5, 6, 7, 10		;REPORT ALL ERRORS
7870								
7871	025362	004437	040776		JSR	R4,GENCOM		;GENERATE DATA OF ZEROS
7872	025366	000001			1			
7873	025370	000400			400			
7874								
7875	025372	004437	034574		JSR	R4,LRLOAD		;LOAD "L" REGS
7876	025376	000123			WRDATA			;WRDATA
7877	025400	177630			-150			; -150 WORDS
7878	025402	062604			OBUFF			;OBUFF IS BUFF ADDRESS
7879	025404	000			.BYTE	0		;SECTOR 0
7880	025405	000			.BYTE	0		;TRACK 0
7881	025406	000000			0			;CYLINDER 0
7882								
7883	025410	104417			TLOADRK			;START OPERATION
7884								
7885	025412	005737	001662	1\$:	TST	INTSET		;CHECK IF INTERRUPT HAS OCCURRED
7886	025416	001026			BNE	2\$;YES - MUCH TO SOON. REPORT ERROR
7887	025420	005762	000002		TST	RKWC(R2)		;TEST IF NPR'S DONE
7888	025424	001372			BNE	1\$;NO - LOOP
7889								
7890	025426	052762	100000 000000		BIS	#CLR,RKCS1(R2)		;CLEAR CONTROLLER (CROWBAR WRITE)
7891								
7892	025434	004437	034574		JSR	R4,LRLOAD		;LOAD "L" REGS
7893	025440	000121			RDDATA			;RDDATA
7894	025442	177400			-400			; -400 WORDS
7895	025444	060604			IBUFF			;IBUFF IS BUFF ADDRESS
7896	025446	000			.BYTE	0		;SECTOR 0
7897	025447	000			.BYTE	0		;TRACK 0
7898	025450	000000			0			;CYLINDER 0
7899								
7900	025452	104417			TLOADRK			;LOAD RK REGS
7901	025454	104425			TWAT48			;WAIT FOR INTERRUPT
7902	025456	104002			ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7903								
7904	025460	104422			TCHKWE			;CHECK OPERATION WITH ERROR
7905	025462	000000			0			
7906	025464	000003			DCKERR!ECHERR			;DATA CHECK AND ECC HARD
7907	025466	000000			0			
7908	025470	104004			ERROR	4 ;OR 5,6,7		;REPORT ALL DISCREPANCIES
7909								
7910	025472	000402			BR	3\$;SKIP TO EXIT
7911	025474			2\$:				

E12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 147
T100 ECC HARD

SEQ 0147

7912 025474 104421
7913 025476 104004
7914 025500

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

3%:

7915
7916
7917
7918
7919
7920
7921
7922
7923
7924
7925
7926

*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
* TRANSITIONS ON DATA LINE.

7927 025500 000004
7928 025502 012737 000012 001262
7929

TEST101: SCOPE
MOV #10.,\$TIMES ;;DO 10. ITERATIONS

7930 025510 104416
7931 025512 104003
7932
7933 025514 004437 034574
7934 025520 000117
7935 025522 000000
7936 025524 000000
7937 025526 000
7938 025527 000
7939 025530 000632

TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REGS
SEEK ;SEEK
0 ;0 WORDS
0 ;0 IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
632 ;CYLINDER 632

7940
7941 025532 104417
7942 025534 104423
7943 025536 104002
7944 025540 005037 001662
7945 025544 104430
7946 025546 104002
7947

TLOADRK ;LOAD RK REGS
TWAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
CLR INTSET ;CLEAR INT FLAG
TWAT96 ;WAIT FOR SECOND INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

7948 025550 004437 034574
7949 025554 000113
7950 025556 000000
7951 025560 000000
7952 025562 000
7953 025563 000
7954 025564 000000
7955

JSR R4,LRLOAD ;LOAD "L" REGS
RECAL ;RECAL
0 ;0 WORDS
0 ;0 IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0

7956 025566 104417
7957 025570 104423
7958 025572 104002
7959

TLOADRK ;LOAD RK REGS
TWAT16 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

7960 025574 004437 034574
7961 025600 000125
7962 025602 000000
7963 025604 000000
7964 025606 000
7965 025607 000
7966 025610 000000
7967 025612 012737 000040 001616

JSR R4,LRLOAD ;LOAD "L" REGS
RDHEAD ;RDHEAD
0 ;0 WORDS
0 ;0 IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0
MOV #DMD,L.MR1 ;SET DIAG MODE


```

7968 025620 104417          TLOADRK          ;LOAD RK REGS
7969
7970 025622 004437 035050   JSR      R4,MCLOCK ;CLOCK CONTROLLER THROUGH SEEK
7971 025626 001062          1062           ;AND CLEAR TO READ
7972
7973 025630 005062 000026   CLR      RKMRI(R2) ;RESET DIAG MODE, LET RD HDRS COMPLETE
7974
7975 025634 104424          TWAT32          ;WAIT FOR INTERRUPT
7976 025636 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
7977 025640 104422          TCHKWE         ;CHECK OPERATION WITH EXP ERROR
7978 025642 010000          DTERR         ;DRIVE TIMING ERROR
7979 025644 000000          0
7980 025646 000000          0
7981 025650 104004          ERROR 4 ;OR 5,6,7 ;REPORT ALL DISCREPANCIES
7982
7983 025652          15:
7984 025652 104416          TSSINIT        ;CLEAR SUBSYSTEM
7985 025654 104003          ERROR 3        ;BAD INIT ERROR
7986 025656 012762 000100 000000  MOV      #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
7987 025664 005037 001662          CLR      INTSET  ;CLEAR INT FLAG
7988
7989 025670 104437          TWAT85        ;WAIT FOR INTERRUPT FOR END OF RECAL
7990 025672 104002          ERROR 2
7991
7992 025674 004437 034574   JSR      R4,LRLOAD ;LOAD "L" REGS
7993 025700 000105          CLEAR        ;CLEAR
7994 025702 000000          0           ;0 WORDS
7995 025704 000000          0           ;0 IS BUFF ADDRESS
7996 025706 000          .BYTE 0       ;SECTOR 0
7997 025707 000          .BYTE 0       ;TRACK 0
7998 025710 000000          0           ;CYLINDER 0
7999
8000 025712 104417          TLOADRK        ;LOAD RK REGS
8001 025714 104423          TWAT16        ;WAIT FOR INTERRUPT
8002 025716 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8003 025720 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
8004 025722 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS

```

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

```

```

8018
8019 025724 000004          ST102: SCOPE
8020 025726 012737 000062 001262  MOV      #50.,$TIMES ;DO 50. ITERATIONS
8021 025734 104416          TSSINIT        ;CLEAR SUBSYSTEM
8022 025736 104003          ERROR 3        ;BAD INIT ERROR
8023

```



```

8024 025740 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
8025 025744 000101 SELDRV ;SELDV
8026 025746 000000 0 ;0 WORDS
8027 025750 000000 0 ;0 IS BUFF ADDRESS
8028 025752 000 .BYTE 0 ;SECTOR 0
8029 025753 000 .BYTE 0 ;TRACK 0
8030 025754 000000 0 ;CYLINDER 0
8031
8032 025756 104417 TLOADRK ;LOAD RK REGS
8033 025760 104423 TWAT16 ;WAIT FOR INTERRUPT
8034 025762 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8035
8036 025764 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8037 025766 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8038
8039 025770 104416 TSSINIT ;CLEAR SUBSYSTEM
8040 025772 104003 ERROR 3 ;BAD INIT ERROR
8041
8042 025774 004437 034574 JSR R4,LRLOAD ;LOAD "L" REGS
8043 026000 000101 SELDRV ;SELDV
8044 026002 000000 0 ;0 WORDS
8045 026004 000000 0 ;0 IS BUFF ADDRESS
8046 026006 000 .BYTE 0 ;SECTOR 0
8047 026007 000 .BYTE 0 ;TRACK 0
8048 026010 000000 0 ;CYLINDER 0
8049 026012 012737 000043 001616 MOV #3!DMD,L.MR1 ;SET DIAG MODE AND MESSAGE PAIR 3
8050 026020 005037 001610 CLR L.CS2 ;SELECT DRIVE 0
8051
8052 026024 104417 TLOADRK ;LOAD RK REGS
8053
8054 026026 004437 035050 JSR R4,MCLOCK ;CLOCK THROUGH PHASE ADDRESS 6
8055 026032 001027 1027
8056
8057 026034 042762 000040 000026 BIC #DMD,RKMR1(R2) ;CLEAR MAINTENANCE MODE
8058
8059 026042 104424 TWAT32 ;WAIT FOR INTERRUPT
8060 026044 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
8061
8062 026046 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8063 026050 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8064
8065 *****
8066 *TEST 103 DRIVE OFF TRACK
8067 *
8068 * ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
8069 * OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC
8070 * MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8071 * TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR
8072 * MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK
8073 * SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILABLE DRIVES.
8074 *
8075 *****
8076 TST103: SCOPE
8077 MOV #10.,$TIMES ;;DO 10. ITERATIONS
8078 TSSINIT ;CLEAR SUBSYSTEM
8079 ERROR 3 ;BAD INIT ERROR

```


8080	026066	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
8081	026072	000113		RECAL		;RECAL
8082	026074	000000		0		;0 WORDS
8083	026076	000000		0		;0 IS BUFF ADDRESS
8084	026100	000		.BYTE	0	;SECTOR 0
8085	026101	000		.BYTE	0	;TRACK 0
8086	026102	000000		0		;CYLINDER 0
8087						
8088	026104	104417		TLOADRK		;LOAD RK REGS
8089	026106	104423		TWAT16		;WAIT FOR INTERRUPT
8090	026110	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8091	026112	005037	001662	CLR	INTSET	;CLEAR INTERRUPT FLAG
8092						
8093	026116	104437		TWAT3S		;WAIT FOR INTERRUPT #2
8094	026120	104002		ERROR	2	
8095						
8096	026122	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
8097	026126	000105		CLEAR		;CLEAR
8098	026130	000000		0		;0 WORDS
8099	026132	000000		0		;0 IS BUFF ADDRESS
8100	026134	000		.BYTE	0	;SECTOR 0
8101	026135	000		.BYTE	0	;TRACK 0
8102	026136	000000		0		;CYLINDER 0
8103						
8104	026140	104417		TLOADRK		;LOAD RK REGS
8105	026142	104423		TWAT16		;WAIT FOR INTERRUPT
8106	026144	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8107						
8108	026146	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8109	026150	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8110						
8111	026152	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS
8112	026156	000115		OFFSET		;OFFSET
8113	026160	000000		0		;0 WORDS
8114	026162	000000		0		;0 IS BUFF ADDRESS
8115	026164	000		.BYTE	0	;SECTOR 0
8116	026165	000		.BYTE	0	;TRACK 0
8117	026166	000000		0		;CYLINDER 0
8118	026170	112737	000060 001612	MOVB	#60,L.ASOF	;SET OFFSET AT +1200
8119						
8120	026176	104417		TLOADRK		;LOAD RK REGS
8121	026200	104423		TWAT16		;WAIT FOR INTERRUPT
8122	026202	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8123						
8124	026204	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8125	026206	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8126						
8127	026210	005037	001662	CLR	INTSET	;CLEAR INT FLAG
8128						
8129	026214	104424		TWAT32		;WAIT FOR INT #2
8130	026216	104002		ERROR	2	
8131						
8132	026220	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8133	026222	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8134						
8135	026224	004437	034574	JSR	R4,LRLOAD	;LOAD "L" REGS


```

8136 026230 000105 CLEAR ;CLEAR
8137 026232 000000 0 ;0 WORDS
8138 026234 000000 0 ;0 IS BUFF ADDRESS
8139 026236 000 .BYTE 0 ;SECTOR 0
8140 026237 000 .BYTE 0 ;TRACK 0
8141 026240 000000 0 ;CYLINDER 0
8142
8143 026242 104417 TLOADRK ;LOAD RK REGS
8144 026244 104423 TWAT16 ;WAIT FOR INTERRUPT
8145 026246 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8146
8147 026250 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8148 026252 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8149
8150 026254 004437 034574 JSR R4,LRLoad ;LOAD "L" REGS
8151 026260 000123 WRDATA ;WRDATA
8152 026262 177777 -1 ;-1 WORDS
8153 026264 062604 OBUFF ;OBUFF IS BUFF ADDRESS
8154 026266 000 .BYTE 0 ;SECTOR 0
8155 026267 000 .BYTE 0 ;TRACK 0
8156 026270 000000 0 ;CYLINDER 0
8157 026272 012737 000040 001616 MOV #DMD,L.MR1 ;SET DIAGNOSTIC MODE
8158
8159 026300 104417 TLOADRK
8160
8161 026302 004437 035050 JSR R4,MCLOCK ;CLOCK THROUGH SEEK & DRIVE CLEAR
8162 026306 001064 1064
8163
8164 026310 005062 000026 CLR RKMR1(R2) ;CLEAR DIAGNOSTIC MODE
8165 026314 104424 TWAT32 ;WAIT FOR INTERRUPT
8166 026316 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
8167
8168 ;:*****
8169 ;:*****
8170
8171 ; NOTE:
8172 ;
8173 ; THE DRIVE LOGIC DOES NOT RAISE ATTENTION AT THE OCCURRENCE
8174 ; OF THE ERROR. IT FIRST RETURNS THE HEADS TO CENTERLINE
8175 ; AND FIRES A HEADS SETTling ONE-SHOT. ATTENTION IS NOT SET
8176 ; UNTIL THE ONE-SHOT TIMES OUT. CONSEQUENTLY THE CONTROLLER
8177 ; WILL FINISH THE WRITE, RAISE INTERRUPT AS THOUGH NO ERROR
8178 ; OCCURRED, THE DRIVE ATTENTION WILL HAPPEN A LITTLE LATER
8179 ; (ABOUT 3 MILLISECONDs) AT WHICH TIME THE DRIVE MUST BE SELECTED
8180 ; TO CHECK THAT DRIVE OFF TRACK SET.
8181
8182 ;:*****
8183 ;:*****
8184
8185 ; THIS DRIVE LOGIC MAY CHANGE!!! IF IT DOES THIS
8186 ; TEST MUST BE ALTERED.
8187
8188 026320 005037 001662 CLR INTSET ;CLEAR INT FLAG
8189 026324 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8190 026326 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8191

```



```

8192 026330 104423          TWAT16          ;WAIT FOR INTERRUPT FROM DRIVE
8193                                ;THAT SIGNALSDRIVE OFF TRACK ERROR
8194 026332 104002          ERROR 2
8195
8196 026334 004437 034574    JSR      R4,LRLOAD      ;LOAD "L" REGS
8197 026340 000101          SELDRV          ;SELDV
8198 026342 000000          0              ;0 WORDS
8199 026344 000000          0              ;0 IS BUFF ADDRESS
8200 026346 000          .BYTE 0          ;SECTOR 0
8201 026347 000          .BYTE 0          ;TRACK 0
8202 026350 000000          0              ;CYLINDER 0
8203 026352 005037 001616    CLR      L.MR1        ;RESET DIAG MODE
8204
8205 026356 104417          TLOADRK        ;LOAD RK REGS
8206 026360 104423          TWAT16        ;WAIT FOR INTERRUPT
8207 026362 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8208
8209 026364 104422          TCHKWE        ;CHECK OPERATION WITH ERROR EXPECTED
8210 026366 000400          DROTERR       ;DRIVE OFF TRACK
8211 026370 000000          0
8212 026372 000000          0
8213 026374 104004          ERROR 4; OR 5,6,7 ;REPORT ANY DISCREPANCIES
8214
8215 *****
8216 *TEST 104      FILE UNSAFE
8217 *
8218 *      ISSUE A SUBSYSTEM CLEAR.  ISSUE A RECLAIBRATE.  ISSUE
8219 *      A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR
8220 *      FORMAT.  DO A SELECT COMMAND IN 26 SECTOR FORMAT.
8221 *      PUT CONTROLLER IN DIAGNOSTIC MODE.  ISSUE A WRITE
8222 *      HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR
8223 *      FORMAT.  CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.
8224 *      SIMULATE INDEX PULSE.  TURN OFF DIAGNOSTIC MODE.  FILE
8225 *      UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE
8226 *      THROUGH SECTOR PULSE.  REPEAT FOR ALL AVAILABLE DRIVES.
8227 *****
8228 026376 000004          TST104: SCOPE
8229 026400 012737 000012 001262  MOV      #10,$TIMES    ;DO 10. ITERATIONS
8230 026406 012737 177777 001664  MOV      #-1,REFMT     ;SET REFORMAT SWITCH
8231 026414 104416          TSSINIT        ;CLEAR SUBSYSTEM
8232 026416 104003          ERROR 3        ;BAD INIT ERROR
8233
8234 026420 004437 034574    JSR      R4,LRLOAD      ;LOAD "L" REGS
8235 026424 000113          RECAL          ;RECAL
8236 026426 000000          0              ;0 WORDS
8237 026430 000000          0              ;0 IS BUFF ADDRESS
8238 026432 000          .BYTE 0          ;SECTOR 0
8239 026433 000          .BYTE 0          ;TRACK 0
8240 026434 000000          0              ;CYLINDER 0
8241
8242 026436 104417          TLOADRK        ;LOAD RK REGS
8243 026440 104423          TWAT16        ;WAIT FOR INTERRUPT
8244 026442 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8245
8246 026444 104421          TCHKOP        ;CHECK OPERATION FOR ANY ERRORS
8247 026446 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS

```


8248							
8249	026450	005037	001662	CLR	INTSET		;CLEAR INT FLAG
8250	026454	104437		TWAT8S			;WAIT FOR SECOND INT
8251	026456	104002		ERROR	2		
8252							
8253	026460	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8254	026462	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8255							
8256	026464	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
8257	026470	000105		CLEAR			;CLEAR
8258	026472	000000		0			;0 WORDS
8259	026474	000000		0			;0 IS BUFF ADDRESS
8260	026476	000		.BYTE	0		;SECTOR 0
8261	026477	000		.BYTE	0		;TRACK 0
8262	026500	000000		0			;CYLINDER 0
8263							
8264	026502	104417		TLOADRK			;LOAD RK REGS
8265	026504	104423		TWAT16			;WAIT FOR INTERRUPT
8266	026506	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8267							
8268	026510	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8269	026512	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8270							
8271	026514	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
8272	026520	010125		RDHEAD!CFMT			;RDHEAD!CFMT
8273	026522	000000		0			;0 WORDS
8274	026524	000000		0			;0 IS BUFF ADDRESS
8275	026526	000		.BYTE	0		;SECTOR 0
8276	026527	000		.BYTE	0		;TRACK 0
8277	026530	000000		0			;CYLINDER 0
8278							
8279	026532	104417		TLOADRK			;LOAD RK REGS
8280	026534	104424		TWAT32			;WAIT FOR INTERRUPT
8281	026536	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8282							
8283	026540	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8284	026542	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
8285							
8286	026544	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
8287	026550	000101		SELDV			;SELDV
8288	026552	000000		0			;0 WORDS
8289	026554	000000		0			;0 IS BUFF ADDRESS
8290	026556	000		.BYTE	0		;SECTOR 0
8291	026557	000		.BYTE	0		;TRACK 0
8292	026560	000000		0			;CYLINDER 0
8293							
8294	026562	104417		TLOADRK			;LOAD RK REGS
8295	026564	104423		TWAT16			;WAIT FOR INTERRUPT
8296	026566	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8297							
8298	026570	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
8299	026574	000127		WRHEAD			;WRHEAD
8300	026576	177777		-1			; -1 WORDS
8301	026600	062604		OBUFF			;OBUFF IS BUFF ADDRESS
8302	026602	000		.BYTE	0		;SECTOR 0
8303	026603	000		.BYTE	0		;TRACK 0


```

8360 026754 012737 000001 001616      MOV      #1,L.MR1      ;SET MESSAGE SELECT ONE
8361
8362 026762      1$:
8363 026762 104417      TLOADRK      ;LOAD RK REGS
8364 026764 104423      TWAT16      ;WAIT FOR INTERRUPT
8365 026766 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8366
8367 026770 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8368 026772 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8369
8370 026774 032737 000040 001574      BIT      #S.HDHM,T.MR2 ;TEST IF HEADS HOME
8371 027002 001767      BEQ      1$
8372
8373 027004 104416      TSSINIT      ;CLEAR SUBSYSTEM
8374 027006 104003      ERROR 3      ;BAD INIT ERROR
8375
8376 027010 005037 001662      CLR      INTSET      ;CLEAR INT FLAG
8377 027014 104434      TWAT159     ;WAIT FOR APPROX 160 MS
8378 027016 000240      NOP      ;DON'T CARE ERROR RETURN
8379
8380 027020 104416      TSSINIT      ;CLEAR SUBSYSTEM
8381 027022 104003      ERROR 3      ;BAD INIT ERROR
8382
8383 027024 012762 000100 000000      MOV      #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
8384
8385 027032 104437      TWAT85      ;WAIT FOR SECOND INTERRUPT
8386 027034 104002      ERROR 2
8387
8388 027036 005037 001616      CLR      L.MR1      ;CLEAR MR1
8389
8390 027042 004437 034574      JSR      R4,LRLOAD   ;LOAD "L" REGS
8391 027046 000105      CLEAR      ;CLEAR
8392 027050 000000      0          ;0 WORDS
8393 027052 000000      0          ;0 IS BUFF ADDRESS
8394 027054 000      .BYTE 0      ;SECTOR 0
8395 027055 000      .BYTE 0      ;TRACK 0
8396 027056 000000      0          ;CYLINDER 0
8397
8398 027060 104417      TLOADRK      ;LOAD RK REGS
8399 027062 104423      TWAT16      ;WAIT FOR INTERRUPT
8400 027064 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8401
8402 027066 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8403 027070 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8404
8405 027072 004437 040776      JSR      R4,GENCOM   ;BUILD HEADERS
8406 027076 001200      1200
8407
8408 027100 004437 034574      JSR      R4,LRLOAD   ;LOAD "L" REGS
8409 027104 000127      WRHEAD      ;WRHEAD
8410 027106 177676      -102       ;-102 WORDS
8411 027110 062604      OBUFF      ;OBUFF IS BUFF ADDRESS
8412 027112 000      .BYTE 0      ;SECTOR 0
8413 027113 000      .BYTE 0      ;TRACK 0
8414 027114 000000      0          ;CYLINDER 0
8415

```



```

8416 027116 104417      TLOADRK          ;LOAD RK REGS
8417 027120 104426      TWAT64           ;WAIT FOR INTERRUPT
8418 027122 104002      ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
8419
8420 027124 104421      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8421 027126 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435 027130 000004      TST105: SCOPE
8436 027132 012737 000001 001262      MOV #1,$TIMES    ;;DO 1 ITERATION
8437
8438 027140 104416      TSSINIT         ;CLEAR SUBSYSTEM
8439 027142 104003      ERROR 3         ;BAD INIT ERROR
8440
8441 027144 013762 001626 000010      MOV DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
8442 027152 012762 000001 000000      MOV #1,RKCS1(R2)   ;SELECT THE DRIVE
8443 027160 032762 000200 000012 1$: BIT #DRDY,RKDS(R2) ;TEST IF DRIVE READY
8444 027166 001774      BEQ 1$          ;NO LOOP
8445
8446 027170 104416      TSSINIT         ;CLEAR SUBSYSTEM
8447 027172 104003      ERROR 3         ;BAD INIT ERROR
8448
8449 027174 004437 040776      JSR R4,GENCOM    ;GENERATE HEADERS FOR CYL 0
8450 027200 001200      1200
8451
8452 027202 004437 034574      JSR R4,LRLOAD    ;LOAD "L" REGS
8453 027206 000127      WRHEAD         ;WRHEAD
8454 027210 177676      -102          ;-102 WORDS
8455 027212 062604      OBUF          ;OBUF IS BUFF ADDRESS
8456 027214 000      .BYTE 0        ;SECTOR 0
8457 027215 000      .BYTE 0        ;TRACK 0
8458 027216 000000      0             ;CYLINDER 0
8459
8460 027220 104417      TLOADRK          ;LOAD RK REGS
8461 027222 104426      TWAT64           ;WAIT FOR INTERRUPT
8462 027224 104002      ERROR 2         ;TO SLOW/NOT COMPLETE ERROR
8463
8464 027226 104421      TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8465 027230 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8466 027232 005037 001664      CLR REFMT       ;CLEAR REFORMAT SWITCH

```



```

0467          027236 000004
0468          027240 012737 000001 001262
0469          027246 032737 000200 001630
0470          027254 001022
0471
0472
0473          027256 005237 001626 1$: INC DRVNUM ;BUMP TO NEXT SEQUENTIAL ADDRESS
0474          027262 006337 001630 ASL DRVBIT ;BUMP DRIVEBIT TO THAT POSITION
0475          027266 033737 001630 001354 BIT DRVBIT,$DEV ;IS THIS DRIVE TO BE TESTED?
0476          027274 001005 BNE 2$ ;YES-EXIT
0477          027276 032737 000400 001630 BIT #BIT8,DRVBIT ;ALL DRIVES TESTED?
0478          027304 001006 BNE 3$ ;YES-EXIT
0479          027306 000763 BR 1$ ;ELSE CHECK NEXT DRIVE AVAILABLE
0480
0481          027310 112737 000004 001102 2$: MOVB #4,$STSTNM ;SET TEST NUMBER FOR REPORTS
0482          027316 000137 004450 JMP TSTLUP ;GO TO TEST LOOP TO CHECK THIS DRIVE
0483          027322 005037 001630 3$: CLR DRVBIT ;CLEAR DRIVE BIT
0484          027326 005037 001626 CLR DRVNUM ;CLEAR DRIVE NUMBER
0485
0486          ::*****
0487          ;*TEST 106 RESET ATTENTIONS WITH UNIBUS INIT
0488          ;*
0489          ;* DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
0490          ;* ISSUE A RESET. MAKE SURE ALL ATTENTION RESET.
0491          ;*
0492          ::*****
0493          TST106: SCOPE
0494          027332 000004 MOV #10,$TIMES ;DO 10. ITERATIONS
0495          027334 012737 000012 001262 CLR R0 ;CLEAR DRIVE POSITION COUNTER
0496          027342 005000 MOV #1,R1 ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
0497          027344 012701 000001 MOV $DEV,R3 ;GET DEVICE MAP
0498          027350 013703 001354 TSSINIT ;CLEAR SUBSYSTEM
0499          027354 104416 ERROR 3 ;BAD INIT ERROR
0500          027360 030103 1$: BIT R1,R3 ;TEST IF THIS DRIVE AVAILABLE
0501          027362 001006 BNE 2$ ;YES-SKIP TO SEEK
0502          027364 006301 3$: ASL R1 ;SHIFT DRIVE SELECT BIT
0503          027366 005200 INC R0 ;BUMP DRIVE POSITION COUNTER
0504          027370 032701 000400 BIT #BIT8,R1 ;ALL DRIVE POSITIONS CHECKED
0505          027374 001771 BEQ 1$ ;NO-LOOP
0506          027376 000441 BR 4$ ;SKIP TO RESET
0507
0508          027400 010037 001610 2$: MOV R0,L,CS2 ;LOAD DRIVE NUMBER
0509          027404 012737 000113 001600 MOV #RECAL,L,CS1 ;LOAD RECALIBRATE
0510
0511          027412 104417 TLOADRK ;LOAD RK REGS
0512          027414 104423 TWAT16 ;WAIT FOR INTERRUPT
0513          027416 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
0514
0515          027420 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
0516          027424 012705 000764 MOV #500,R5 ;SET COUNT FOR 8 SECONDS
0517          027430 012762 000012 000000 MOV #12,RKCS1(R2) ;RESET INTERRUPT ENABLE
0518          027436 016237 000016 001556 12$: MOV RKA$OF(R2),T,ASOF ;GET ATTENTION REGISTER
0519          027444 113704 001557 MOV T,ASOF+1,R4 ;ADJUST FOR CHECK OF ATTENTION
0520          027450 042704 177400 MOVB #177400,R4 ;CLEAR UNUSED BITS
0521          027454 030104 BIT R1,R4 ;CHECK IF ATT SET FROM DRIVE RECAL'ED
0522          027456 001006 BNE 10$ ;YES - SKIP

```


C13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 158
T106 RESET ATTENTIONS WITH UNIBUS INIT

SEQ 0158

```

8523
8524 027460 104423          TWAT16          ;WAIT FOR 16 MS
8525 027462 000240          NOP             ;DON'T CARE RETURNS
8526 027464 000240          NOP
8527 027466 005305          DEC            R5          ;TATOL WAIT TIME IS 8 SECONDS
8528 027470 001362          BNE            12$         ;CHECK ATTENTION EACH 16 MS
8529 027472 104002          ERROR         2          ;REPORT IF NO ATTENTION IN 8 SEC
8530
8531 027474          10$:          TCHKOP
8532 027474 104421          ERROR         4 ;OR 5, 6, 7 ;CHECK OPERATION FOR ANY ERRORS
8533 027476 104004          ;REPORT ALL ERRORS
8534
8535 027500 000731          BR             3$          ;LOOP FOR NEXT DRIVE
8536 027502 000005          4$:          RESET          ;UNIBUS RESET
8537 027504 004737 043664          JSR            PC,$TKINT  ;RESET KEYBOARD INTERRUPT ENABLE
8538
8539 027510 012701 000031          MOV            #25.,R1     ;DO A SHORT DELAY
8540 027514 005301          5$:          DEC            R1
8541 027516 001376          BNE            5$
8542 027520 004737 033704          JSR            PC,OPTTST  ;SET UP OPTIONS
8543
8544 027524 104420          TGETRK        ;GET RK611 REGS
8545
8546 027526 105737 001557          TSTB          T,ASOF+1    ;ALL ATTENTION RESET?
8547 027532 001407          BEQ            6$          ;YES-SKIP
8548
8549 027534 012737 055421 001470          MOV            #EMDA,EM12N ;"DRIVE ATT NOT RESET RESULT OF
8550 027542 012737 055545 057702          MOV            #EMUR,DF011A ;UNIBUS RESET"
8551 027550 104012          ERROR         12
8552
8553 027552          6$:
8554          ;*****
8555          ;*TEST 107      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
8556          ;*
8557          ;*      DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
8558          ;*      ISSUE A SUBSYSTEM CLEAR.  MAKE SURE ALL ATTENTIONS
8559          ;*      RESET.
8560          ;*
8561          ;*****
8562 027552 000004          †ST107: SCOPE
8563 027554 012737 000062 001262          MOV            #50.,$TIMES ;DO 50. ITERATIONS
8564 027562 005000          CLR            R0         ;CLEAR DRIVE POSITION COUNTER
8565 027564 012701 000001          MOV            #1,R1      ;PRESET TO TEST POSITION 0
8566 027570 013703 001354          MOV            $DEVN,R3   ;CUT DEVICE MAP
8567 027574 104416          TSSINIT
8568 027576 104003          ERROR         3          ;CLEAR SUBSYSTEM
8569 027600 030103          1$:          BIT            R1,R3     ;BAD INIT ERROR
8570 027602 001006          BNE            2$         ;THIS DRIVE AVAILABLE?
8571 027604 006301          3$:          ASL            R1         ;YES-SKIP TO SEEK
8572 027606 005200          INC            R0         ;SHIFT TO NEXT DRIVE POSITION
8573 027610 032701 000400          BIT            #BIT8,R1   ;DUMP POSITION COUNTER
8574 027614 001771          BEQ            1$         ;ALL POSITIONS CHECKED
8575 027616 000441          BR             4$         ;NO-LOOP
8576
8577 027620 010037 001610          2$:          MOV            R0,L,CS2   ;LOAD DRIVE NUMBER
8578 027624 012737 000113 001600          MOV            #RECAL,L,CS1 ;LOAD RECALIBRATE

```


D13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 159
T107 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

SEQ 0159

```

8579 027632 104417 TLOADRK ;LOAD RK REGS
8580 027634 104423 TWAT16 ;WAIT FOR INTERRUPT
8581 027636 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8582
8583 027640 005037 001662 CLR INTSET ;CLEAR INT FLAG
8584 027644 012705 000764 MOV #500,R5 ;SET COUNT FOR 8 SECONDS
8585 027650 012762 000012 000000 MOV #12,RKCS1(R2) ;RESET INTERRUPT ENABLE
8586 027656 016237 000016 001556 12$: MOV RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8587 027664 113704 001557 MOV T.ASOF+1,R4 ;ADJUST FOR CHECK OF ATTENTION
8588 027670 042704 177400 BIC #177400,R4 ;CLEAR UNUSED BITS
8589 027674 030104 BIT R1,R4 ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8590 027676 001006 BNE 10$ ;YES - SKIP
8591
8592 027700 104423 TWAT16 ;WAIT FOR 16 MS
8593 027702 000240 NOP ;DON'T CARE RETURNS
8594 027704 000240 NOP
8595 027706 005305 DEC R5 ;TATOL WAIT TIME IS 8 SECONDS
8596 027710 001362 BNE 12$ ;CHECK ATTENTION EACH 16 MS
8597 027712 104002 ERROR 2 ;REPORT IF NO ATTENTION IN 8 SEC
8598
8599 027714 10$:
8600 027714 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8601 027716 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8602
8603 027720 000731 BR 3$ ;LOOP FOR NEXT DRIVE
8604
8605 027722 052762 000040 000010 4$: BIS #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
8606 027730 012701 000031 MOV #25.,R1 ;DO A SHORT DELAY
8607 027734 005301 5$: DEC R1
8608 027736 001376 BNE 5$
8609
8610 027740 104420 TGETRK ;GET RK611 REGS
8611
8612 027742 105737 001557 TSTB T.ASOF+1 ;TEST ALL ATTENTION RESET
8613 027746 001407 BEQ 6$ ;YES-SKIP
8614
8615 027750 012737 055421 001470 MOV #EMDA,EM12N ;"DRIVE ATT NOT RESET AS RESULT OF
8616 027756 012737 055635 057702 MOV #EMSCLR,DF011A ;SUBSYSTEM CLEAR"
8617 027764 104012 ERROR 12
8618
8619 027766 6$:
8620 ;*****
8621 ;*TEST 110 SVAL AND ATTENTION FROM OTHER DRIVE
8622 ;*
8623 ;* DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT
8624 ;* ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID
8625 ;* IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
8626 ;* MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
8627 ;* CHANGE REMAINS RESET.
8628 ;*
8629 ;* REPEAT FOR ALL COMBINATIONS OF TWO AVAILIABLE DRIVES.
8630 ;*
8631 ;* NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST
8632 ;* TWO DRIVES ARE AVAILABLE.
8633 ;*
8634 ;*****

```


E13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 160
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0160

```

8635 027766 000004          TST110: SCOPE
8636 027770 012737 000024 001262  MOV      #20, $TIMES      ;; DO 20. ITERATIONS
8637 027776 013746 001354          MOV      $DEVN, -(SP)    ;; PUT DEVICE MAP ON STACK
8638 030002 004437 034774          JSR      R4, BITCNT     ;; COUNT NUMBER OF BITS (# OF DRIVES)
8639 030006 022627 000001          CMP      (SP)+, #1      ;; COMPARE TO 1
8640 030012 101007          BHI     2$              ;; SKIP IF MORE THAN 1
8641 030014 005737 001304          TST     $PASS          ;; CHECK IF PASS 0
8642 030020 001002          BNE     1$              ;; NO-SKIP
8643
8644 030022 104401 051477          TYPE    ,OPR018        ;; "OVERLAPPED OPERATION BYPASSED"
8645 030026 000137 031242          1$:     JMP      $EOP        ;; GET OUT.
8646
8647 030032 012737 177777 001224  2$:     MOV      #-1, $TMP1     ;; SET LOOP CONTROL FLAG
8648 030040 013705 001354          MOV      $DEVN, R5     ;; GET DEVICE MAP
8649 030044 005000          CLR     R0              ;; CLEAR FOR DRIVE #A
8650 030046 005001          CLR     R1              ;; CLEAR FOR DRIVE #B
8651 030050 012703 000001          MOV      #1, R3        ;; SET DRIVE POSITION A
8652 030054 012704 000001          MOV      #1, R4        ;; SET DRIVE POSITION B
8653 030060 012737 030070 001110  MOV      #3$, $LPERR    ;; SET LOCAL LOOP ON ERROR
8654 030066 000477          BR      11$            ;; GO SET UP POINTERS
8655
8656 030070          3$:
8657 030070 104416          TSSINIT          ;; CLEAR SUBSYSTEM
8658 030072 104003          ERROR  3          ;; BAD INIT ERROR
8659
8660 030074 010037 001610          MOV      R0, L.CS2     ;; LOAD DRIVE A
8661 030100 010037 001626          MOV      R0, DRVNUM    ;; LOAD FOR REPORT
8662 030104 012737 000113 001600  MOV      #RECAL, L.CS1 ;; LOAD RECAL
8663
8664 030112 104417          TLOADRK          ;; LOAD RK REGS
8665 030114 104423          TWAT16          ;; WAIT FOR INTERRUPT
8666 030116 104002          ERROR  2          ;; TO SLOW/NOT COMPLETE ERROR
8667 030120 104421          TCHKOP          ;; CHECK OPERATION FOR ANY ERRORS
8668 030122 104004          ERROR  4 ;OR 5, 6, 7 ;; REPORT ALL ERRORS
8669
8670 030124 005037 001662          CLR     INTSET        ;; CLEAR INT FLAG
8671
8672 030130 010137 001610          MOV      R1, L.CS2     ;; LOAD DRIVE B
8673 030134 010137 001626          MOV      R1, DRVNUM    ;; LOAD FOR REPORT
8674 030140 012737 000101 001600  MOV      #SELDRV, L.CS1 ;; LOAD DRIVE SELECT
8675
8676 030146 104417          TLOADRK          ;; LOAD RK REGS
8677 030150 104423          TWAT16          ;; WAIT FOR INTERRUPT
8678 030152 104002          ERROR  2          ;; TO SLOW/NOT COMPLETE ERROR
8679
8680 030154 104421          TCHKOP          ;; CHECK OPERATION FOR ANY ERRORS
8681 030156 104004          ERROR  4 ;OR 5, 6, 7 ;; REPORT ALL ERRORS
8682
8683 030160 032737 100000 001552  BIT      #SVAL, T.DS    ;; CHECK IF STATUS VALID SET
8684 030166 001007          BNE     4$            ;; YES - SKIP
8685 030170 012737 055235 001460  MOV      #EMSVAL, EM11N ;; "STATUS VALID NOT SET RESULT OF
8686 030176 012737 046676 057702  MOV      #OPER00, DF011A ;; DRIVE SELECT"
8687 030204 104011          ERROR  11
8688
8689 030206 005037 001662          4$:     CLR     INTSET        ;; CLEAR INT FLAG
8690 030212 104436          TWAT2S          ;; WAIT FOR SEEK COMPLETE INTERRUPT

```


F13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 161
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0161

```

8691 030214 000401 BR 44$ ;NONE RECEIVED - SKIP
8692 030216 000406 BR 55$ ; RECEIVED - SKIP
8693
8694 030220 010037 001626 44$: MOV R0,DRVNUM ;SET DRIVE FOR REPORT
8695 030224 012737 057272 001372 MOV #DHO17,DH2N ;"COMMAND - SELECT AFTER RECAL"
8696 030232 104002 ERROR 2
8697
8698 030234 104420 55$: TGETRK ;GET RK REGS
8699 030236 032737 100000 001552 BIT #SVAL,T.DS ;TEST IF SVAL STILL SET
8700 030244 001007 BNE 5$ ;YES - SKIP
8701
8702 030246 012737 055235 001510 MOV #EMSVAL,EM14N ;"STATUS VALID RESET RESULT OF
8703 030254 012737 057315 057702 MOV #DHO18,DF011A ;RECAL COMPLETE ATTENTION AFTER SEL"
8704 030262 104014 ERROR 14
8705
8706 030264 104415 5$: SCOP1 ;LOCAL LOOP TO 3$
8707
8708 :
8709 : THE FOLLOWING CODE CAUSES THE TEST TO BE RUN ON EVERY COMBINATION
8710 : OF DRIVES AVAILABLE. THE FIRST PASS OF THE PROGRAM WILL USE THE
8711 : LOWEST NUMBER DRIVE AS A AND THE NEXT HIGHER NUMBER DRIVE AS
8712 : B. THE SECOND PASS SWAPS DRIVE A & B. THE THIRD PASS USES
8713 : THE LOWEST NUMBER DRIVE AS B AND THE 3RD HIGHEST NUMBER DRIVE
8714 : AS A. THE FORTH PASS SWAPS A & B AGAIN. THIS CONTINUES
8715 : UNTIL ALL DRIVES HAVE BEEN TESTED WITH THE LOWEST NUMBER
8716 : DRIVE.
8717 :
8718 : THE SECOND HIGHEST NUMBER DRIVE IS THEN USED AS A AND THE
8719 : THIRD HIGHEST AS B. THEY ARE SWAPPED ON THE NEXT PASS.
8720 :
8721 : THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE
8722 : CHECKED.
8723 030266 005237 001224 11$: INC $TMP1 ;INCREMENT PASS CONTROL
8724 030272 001024 BNE 16$ ;SKIP IF NOT ZERO
8725 : ;(IT WILL BE ZERO ON THE 1ST PASS)
8726
8727 030274 030305 12$: BIT R3,R5 ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8728 030276 001006 BNE 13$ ;YES-SKIP
8729
8730 030300 005200 22$: INC R0 ;BUMP R0 (DRIVE A)
8731 030302 006303 ASL R3 ;SHIFT DRIVE SELECT BIT ONE POSITION
8732 030304 032703 000400 BIT #BIT8,R3 ;IF BIT 8 IS SET, ALL DRIVES HAVE
8733 030310 001771 BEQ 12$ ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8734 030312 000464 BR 50$ ;DONE-EXIT
8735
8736 030314 010001 13$: MOV R0,R1 ;SET DRIVE B TO THE SAME AS A
8737 030316 010304 MOV R3,R4
8738 030320 005201 14$: INC R1 ;BUMP R1 (DRIVE B)
8739 030322 006304 ASL R4 ;SHIFT SELECTOR BIT ONE POSITION
8740 030324 030405 BIT R4,R5 ;IS THIS DRIVE AVAIL?
8741 030326 001004 BNE 15$ ;YES-SKIP
8742 030330 032704 000400 BIT #BIT8,R4 ;WERE ALL POSITIONS CHECKED?
8743 030334 001771 BEQ 14$ ;NO-LOOP
8744 030336 000452 BR 50$ ;DONE-EXIT
8745
8746 030340 000137 030070 15$: JMP 3$ ;GO DO THE TEST ON THE DRIVE A & B

```



```

8747                                     ;CONTAINED IN R0 & R1
8748 030344 032737 000001 001224 16$: BIT  #BIT0,$TMP1 ;IS PASS FLAGS ODD?
8749 030352 001410 BEQ 17$ ;NO-SKIP
8750
8751 030354 010046 MOV R0,-(SP) ;
8752 030356 010346 MOV R3,-(SP) ;SWAP R0 & R1, R3 & R4
8753 030360 010403 MOV R4,R3 ;TO EXCHANGE DRIVE A & B
8754 030362 010100 MOV R1,R0
8755 030364 012604 MOV (SP)+,R4
8756 030366 012601 MOV (SP)+,R1
8757 030370 000137 030070 JMP 3$ ;REPEAT TEST ON THIS COMBO.
8758
8759 030374 032737 000002 001224 17$: BIT  #BIT1,$TMP1 ;TEST IF PASS FLAGS AT HALF MODULE 4?
8760 030402 001410 BEQ 19$ ;NO-SKIP TO BUMP DRIVE B
8761 030404 005200 18$: INC R0 ;BUMP DRIVE A
8762 030406 006303 ASL R3 ;SHIFT DRIVE SELECT BIT
8763 030410 030305 BIT R3,R5 ;AVAILABLE?
8764 030412 001014 BNE 20$ ;YES-SKIP
8765 030414 032703 000400 BIT  #BIT8,R3 ;ALL CHECKED?
8766 030420 001771 BEQ 18$ ;NO-SKIP
8767 030422 000412 BR 21$ ;GO TO NEXT PASS
8768
8769 030424 005201 19$: INC R1 ;BUMP DRIVE B
8770 030426 006304 ASL R4 ;SHIFT DRIVE SELECT BIT
8771 030430 030405 BIT R4,R5 ;AVAILABLE?
8772 030432 001004 BNE 20$ ;YES-SKIP
8773 030434 032704 000400 BIT  #BIT8,R4 ;ALL CHECKED?
8774 030440 001771 BEQ 19$ ;NO-LOOP
8775 030442 000404 BR 23$ ;YES-SKIP TO NEXT PASS
8776
8777 030444 000137 030070 20$: JMP 3$ ;GO TEST THIS COMBO
8778
8779 030450 010100 21$: MOV R1,R0 ;SET DRIVE 0 TO LOW POSITION THIS PASS
8780 030452 010403 MOV R4,R3 ;SET SELECT BITS TO AGREE
8781 030454 005037 001224 23$: CLR $TMP1 ;CLEAR PASS FLAGS
8782 030460 000137 030300 JMP 22$ ;GO SET UP A & B
8783 030464
8784 50$: *****
8785 ;*TEST 111 OVERLAPPED OPERATIONS
8786 ;*
8787 ;* DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A
8788 ;* SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE
8789 ;* DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
8790 ;* AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
8791 ;* BE SET AND DRIVE A HAS ATTENTION SET.
8792 ;*
8793 ;* REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
8794 ;*
8795 ;* NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE
8796 ;* TEST WILL NOT BE DONE.
8797 ;*
8798 ;* *****
8799 030464 000004 TST111: SCOPE
8800 030466 012737 000024 001262 MOV #20,$TIMES ;;DO 20. ITERATIONS
8801
8802 030474 012737 177777 001224 2$: MOV #-1,$TMP1 ;SET LOOP CONTROL FLAG

```


H13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 163
T111 OVERLAPPED OPERATIONS

SEQ 0163

8803	030502	013705	001354		MOV	\$DEVN,R5	;GET DEVICE MAP
8804	030506	005000			CLR	R0	;CLEAR FOR DRIVE #A
8805	030510	005001			CLR	R1	;CLEAR FOR DRIVE #B
8806	030512	012703	000001		MOV	#1,R3	;SET DRIVE POSITION A
8807	030516	012704	000001		MOV	#1,R4	;SET DRIVE POSITION B
8808	030522	012737	030532	001110	MOV	#3\$, \$LPERR	;SET LOCAL LOOP ON ERROR
8809	030530	000545			BR	11\$;GO SET UP POINTERS
8810	030532						
8811	030532	104416			TSSINIT		;CLEAR SUBSYSTEM
8812	030534	104003			ERROR	3	;BAD INIT ERROR
8813							
8814	030536	010037	001626		MOV	R0,DRVNUM	;STORE DRIVE FOR REPORT
8815	030542	010037	001610		MOV	R0,L.CS2	;SETUP DRIVE A TO RECAL
8816	030546	012737	000113	001600	MOV	#RECAL,L.CS1	
8817							
8818	030554	104417			TLOADRK		;LOAD RK REGS
8819	030556	104423			TWAT16		;WAIT FOR INTERRUPT
8820	030560	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8821	030562	005037	001662		CLR	INTSET	;CLEAR INTERRUPT FLAG
8822							
8823	030566	104437			TWAT8S		;WAIT FOR SECOND INTERRUPT
8824	030570	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8825							
8826	030572	012737	000105	001600	MOV	#CLEAR,L.CS1	;SET UP TO CLEAR DRIVE
8827	030600	104417			TLOADRK		;LOAD RK REGS
8828	030602	104423			TWAT16		;WAIT FOR INTERRUPT
8829	030604	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8830	030606						
8831	030606	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8832	030610	104004			ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8833	030612	032737	040000	001540	BIT	#DI,T.CS1	;TEST IF DI STILL SET
8834	030620	001372			BNE	4\$;YES - LOOP
8835	030622	010137	001626		MOV	R1,DRVNUM	;STORE DRIVE FOR REPORT
8836	030626	010137	001610		MOV	R1,L.CS2	;SETUP DRIVE B TO RECAL
8837	030632	012737	000113	001600	MOV	#RECAL,L.CS1	
8838							
8839	030640	104417			TLOADRK		;LOAD RK REGS
8840	030642	104423			TWAT16		;WAIT FOR INTERRUPT
8841	030644	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8842	030646	005037	001662		CLR	INTSET	;CLEAR INTERRUPT FLAG
8843	030652	104437			TWAT8S		;WAIT FOR SECOND INTERRUPT
8844	030654	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8845	030656	012737	000105	001600	MOV	#CLEAR,L.CS1	;SET UP DRIVE CLEAR
8846	030664	104417			TLOADRK		;LOAD RK REGS
8847	030666	104423			TWAT16		;WAIT FOR INTERRUPT
8848	030670	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8849	030672						
8850	030672	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8851	030674	104004			ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8852	030676	032737	040000	001540	BIT	#DI,T.CS1	;TEST IF DI STILL SET
8853	030704	001372			BNE	5\$;YES - LOOP
8854							
8855	030706	010037	001626		MOV	R0,DRVNUM	;STORE DRIVE FOR REPORT
8856	030712	010037	001610		MOV	R0,L.CS2	;SETUP DRIVE A TO SEEK
8857	030716	012737	000001	001614	MOV	#1,L.DCYL	;TO CYL 1
8858	030724	012737	000117	001600	MOV	#SEEK,L.CS1	

3\$:

4\$:

5\$:


```

8859
8860 030732 104417          TLOADRK          ;LOAD RK REGS
8861 030734 104423          TWAT16          ;WAIT FOR INTERRUPT
8862 030736 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8863
8864 030740 010137 001626    MOV R1,DRVNUM   ;STORE DRIVE FOR REPORT
8865 030744 010137 001610    MOV R1,L.CS2   ;SETUP DRIVE B TO WRITE DATA
8866 030750 012737 000100 001614    MOV #100,L.DCYL ;AT CYL 100
8867 030756 012737 177400 001602    MOV #-400,L.WC ;400 WORDS
8868 030764 012737 062604 001604    MOV #OBUF,L.BA
8869 030772 012737 000123 001600    MOV #WRDATA,L.CS1
8870
8871 031000 104417          TLOADRK          ;LOAD RK REGS-DO WRITE
8872
8873 031002 104427          TWAT80          ;WAIT FOR INTERRUPT
8874 031004 104002          ERROR 2        ;TO SLOW/NOT COMPLETE ERROR
8875
8876 031006 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8877 031010 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8878
8879 031012 010037 001626    MOV R0,DRVNUM   ;STORE DRIVE FOR REPORT
8880 031016 130337 001557    BITB R3,T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS DRIVE A
8881 031022 001007          BNE 10$        ;YES-SKIP
8882 031024 012737 055421 001460    MOV #EMDA,EM11N ;"DRIVE ATTENTION NOT SET RESULT OF
8883 031032 012737 047010 057702    MOV #EMSK,DF011A ;SEEK"
8884 031040 104011          ERROR 11
8885
8886 031042 104415          10$: SCOP1      ;LOCAL LOOP TO 3$
8887
8888 :
8889 :
8890 :
8891 :
8892 :
8893 :
8894 :
8895 :
8896 :
8897 :
8898 :
8899 :
8900 :
8901 :
8902 :
8903 031044 005237 001224          11$: INC $TMP1      ;INCREMENT PASS CONTROL
8904 031050 001024          BNE 16$        ;SKIP IF NOT ZERO
8905 :
8906 :
8907 031052 030305          12$: BIT R3,R5   ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8908 031054 001006          BNE 13$        ;YES-SKIP
8909
8910 031056 005200          22$: INC R0      ;BUMP R0 (DRIVE A)
8911 031060 006303          ASL R3         ;SHIFT DRIVE SELECT BIT ONE POSITION
8912 031062 032703 000400    BIT #BIT8,R3  ;IF BIT 8 IS SET, ALL DRIVES HAVE
8913 031066 001771          BEQ 12$        ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8914 031070 000464          BR 50$        ;DONE-EXIT

```



```

8915
8916 031072 010001      13$:  MOV    R0,R1      ;SET DRIVE B TO THE SAME AS A
8917 031074 010304      MOV    R3,R4
8918 031076 005201      14$:  INC    R1        ;BUMP R1 (DRIVE B)
8919 031100 006304      ASL   R4        ;SHIFT SELECTOR BIT ONE POSITION
8920 031102 030405      BIT   R4,R5    ;IS THIS DRIVE AVAIL?
8921 031104 001004      BNE   15$      ;YES-SKIP
8922 031106 032704 000400  BIT   #BIT8,R4 ;WERE ALL POSITIONS CHECKED?
8923 031112 001771      BEQ   14$      ;NO-LOOP
8924 031114 000452      BR    50$     ;DONE-EXIT
8925
8926 031116 000137 030532  15$:  JMP    3$     ;GO DO THE TEST ON THE DRIVE A & B
8927
8928 031122 032737 000001 001224 16$:  BIT   #BIT0,$TMP1 ;CONTAINED IN R0 & R1
8929 031130 001410      BEQ   17$     ;IS PASS FLAGS ODD?
8930
8931 031132 010046      MOV   R0,-(SP) ;
8932 031134 010346      MOV   R3,-(SP) ;SWAP R0 & R1, R3 & R4
8933 031136 010403      MOV   R4,R3   ;TO EXCHANGE DRIVE A & B
8934 031140 010100      MOV   R1,R0
8935 031142 012604      MOV   (SP)+,R4
8936 031144 012601      MOV   (SP)+,R1
8937 031146 000137 030532  JMP    3$     ;REPEAT TEST ON THIS COMBO.
8938
8939 031152 032737 000002 001224 17$:  BIT   #BIT1,$TMP1 ;TEST IF PASS FLAGS AT HALF MODULE 4?
8940 031160 001410      BEQ   19$     ;NO-SKIP TO BUMP DRIVE B
8941 031162 005200      18$:  INC    R0      ;BUMP DRIVE A
8942 031164 006303      ASL   R3      ;SHIFT DRIVE SELECT BIT
8943 031166 030305      BIT   R3,R5   ;AVAILABLE?
8944 031170 001014      BNE   20$     ;YES-SKIP
8945 031172 032703 000400  BIT   #BIT8,R3 ;ALL CHECKED?
8946 031176 001771      BEQ   18$     ;NO-SKIP
8947 031200 000412      BR    21$     ;GO TO NEXT PASS
8948
8949 031202 005201      19$:  INC    R1     ;BUMP DRIVE B
8950 031204 006304      ASL   R4     ;SHIFT DRIVE SELECT BIT
8951 031206 030405      BIT   R4,R5   ;AVAILABLE?
8952 031210 001004      BNE   20$     ;YES-SKIP
8953 031212 032704 000400  BIT   #BIT8,R4 ;ALL CHECKED?
8954 031216 001771      BEQ   19$     ;NO-LOOP
8955 031220 000404      BR    23$     ;YES-SKIP TO NEXT PASS
8956
8957 031222 000137 030532  20$:  JMP    3$     ;GO TEST THIS COMBO
8958
8959 031226 010100      21$:  MOV   R1,R0   ;SET DRIVE 0 TO LOW POSITION THIS PASS
8960 031230 010403      MOV   R4,R3   ;SET SELECT BITS TO AGREE
8961 031232 005037 001224  23$:  CLR   $TMP1   ;CLEAR PASS FLAGS
8962 031236 000137 031056  JMP    22$     ;GO SET UP A & B
8963 031242

```


8964
8965
8966
8967
8968
8969
8970
8971
8972
8973 031242
8974 031242 000004
8975 031244 005037 001102
8976 031250 005037 001262
8977 031254 005237 001304
8978 031260 042737 100000 001304
8979 031266 005327
8980 031270 000001
8981 031272 003063
8982 031274 012737
8983 031276 000001
8984 031300 031270
8985 031302 104401 031310
8986 031306 000407
8987
8988 031326
8989 031326 013746 001304
8990
8991 031332 104405
8992 031334 104401 031342
8993 031340 000421
8994
8995 031404
8996 031404 013746 001112
8997
8998 031410 104405
8999 031412 104401 001273
9000 031416 005037 001112
9001 031422 013700 000042
9002 031426 001405
9003 031430 000005
9004 031432 004710
9005 031434 000240
9006 031436 000240
9007 031440 000240
9008 031442
9009 031442 000137
9010 031444 003116
9011 031446 377 000
9012 031452
9013
9014
9015
9016
9017
9018
9019

```
.SBTTL END OF PASS ROUTINE

;*****
;*INCREMENT THE PASS NUMBER ($PASS)
;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO TST1

$EOP:
SCOPE
CLR $STNM ;:ZERO THE TEST NUMBER
CLR $TIMES ;:ZERO THE NUMBER OF ITERATIONS
INC $PASS ;:INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;:DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;:LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;:YES
MOV (PC)+,2(PC)+ ;:RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT
TYPE ,65$ ;:TYPE ASCIZ STRING
BR ,64$ ;:GET OVER THE ASCIZ
;:65$: .ASCIZ <12><15>/END PASS #/
64$:
MOV $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
;:TYPE PASS NUMBER
;:GO TYPE--DECIMAL ASCII WITH SIGN
TYPE ,67$ ;:TYPE ASCIZ STRING
BR ,66$ ;:GET OVER THE ASCIZ
;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
MOV $ERTTL,-(SP) ;:SAVE $ERTTL FOR TYPEOUT
;:TOTAL NUMBER OF ERRORS
;:GO TYPE--DECIMAL ASCII WITH SIGN
TYPE , $CRLF ;:TYPE CARRIAGE RETURN, LINE FEED
$GET42: MOV 2#42,RO ;:CLEAR ERROR TOTAL
BEQ $DOAGN ;:GET MONITOR ADDRESS
RESET ;:BRANCH IF NO MONITOR
SENDAD: JSR PC,(RO) ;:CLEAR THE WORLD
NOP ;:GO TO MONITOR
NOP ;:SAVE ROOM
NOP ;:FOR
;:ACT11
$DOAGN:
JMP 2(PC)+ ;:RETURN
$RTNAD: .WORD TST1
$ENULL: .BYTE -1,-1,0 ;:NULL CHARACTER STRING
.EVEN
.SBTTL ROUTINE TO SIZE MEMORY

;*****
;*CALL:
;* JSR PC,$SIZE
;* RETURN
;*$LSTAD WILL CONTAIN:
```



```

9020      ;*      WITH KT11 OPTION      -- LAST VIRTUAL ADDRESS OF THE LAST BANK
9021      ;*      WITHOUT KT11 OPTION  -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
9022      ;*SLSTAK WILL CONTAIN THE LAST BANK AS A SAF
9023      ;*SKT11 IS THE MEMORY MANAGEMENT KEY
9024      ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
9025      ;*      MUST BE SETUP BEFORE THE CALL
9026      ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
9027      ;*      DETERMINED BY ROUTINE
9028
9029      $SIZE:  MOV      RO,-(SP)      ;;SAVE RO ON THE STACK
9030      MOV      R1,-(SP)      ;;SAVE R1 ON THE STACK
9031      MOV      R2,-(SP)      ;;SAVE R2 ON THE STACK
9032      MOV      R3,-(SP)      ;;SAVE R3 ON THE STACK
9033      MOV      @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
9034      MOV      @#ERRVEC+2,-(SP)
9035      MOV      SP,RO          ;;SAVE THE STACK POINTER
9036      ;;SET THE ERRVEC PS TO THE PRESENT PS
9037      TRAP
9038      MOV      (SP)+,@#ERRVEC+2 ;;PUSH OLD PSW AND PC ON STACK
9039      MOV      #3776,R1      ;;SAVE THE PSW IN @#ERRVEC+2
9040      TSTB     (PC)+         ;;SETUP ADDRESS
9041      .WORD    200          ;;USE MEMORY MANAGEMENT?
9042      BPL      $SCORE       ;;SET TO USE MEMORY MANAGEMENT
9043      MOV      #SKTNEX,@#ERRVEC ;;BR IF NO
9044      TST      @#SRO         ;;SET FOR TIMEOUT
9045      BIS      #100000,$KT11 ;;KT11 ARE YOU THERE?
9046      CLR      -(SP)        ;;YES--SET KT11 KEY
9047      MOV      #KIPAR0,R2    ;;INITIALIZE FOR "PAR" LOADING
9048      MOV      #↑D8,R3      ;;ADDRESS OF FIRST "PAR"
9049      MOV      #77406,-40(R2) ;;LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
9050      MOV      (SP),(R2)+    ;;PDR = 4K, UP, READ/WRITE
9051      ADD      #200,(SP)     ;;LOAD "PAR"
9052      SOB      R3,1$        ;;UPDATE FOR NEXT "PAR"
9053      MOV      #177600,-(R2) ;;LOOP UNTIL ALL EIGHT ARE LOADED
9054      CLR      -(R2)        ;;SETUP KIPAR7 FOR I/O
9055      MOV      #2$,@#ERRVEC  ;;SETUP KIPAR6 FOR TESTING
9056      MOV      #20,@#SR3    ;;CATCH TIMEOUT IF NO SR3
9057      BR      3$           ;;ENABLE 22 BIT MODE
9058      CMP      (SP)+,(SP)+  ;;THIS PDP-11 HAS A SR3 REGISTER
9059      INC      @#SRO        ;;CLEAN OFF THE STACK--NO SR3
9060      MOV      #SKTOUT,@#ERRVEC ;;TURN ON MEMORY MANAGEMENT
9061      TST      @#143776     ;;SET FOR TIME OUT
9062      ADD      #40,(R2)     ;;TRAP ON NON-EX-MEM
9063      CMP      @#KIPAR7,(R2) ;;MAKE A 1K STEP
9064      BHI      4$         ;;LAST ONE?
9065      MOV      (R2),R2      ;;NO--TRY IT
9066      CLR      @#SRO       ;;GET LAST BANK+1
9067      BR      $SIZEX      ;;TURN OFF MEMORY MANAGEMENT
9068      $SIZEX
9069      BIC      #100000,$KT11 ;;KT11 NON-EXISTENT
9070      MOV      #SCROUT,@#ERRVEC ;;SET FOR TIMEOUT
9071      CLR      R2          ;;SET UP BANK
9072      ADD      #4000,R1     ;;INCREMENT BY 1K
9073      ADD      #40,R2      ;;1K STEP
9074      TST      (R1)        ;;TRAP ON TIME OUT
9075      CMP      #177776,R1  ;;LAST ONE
          BNE      1$        ;;NO--TRY AGAIN

```



```

9076 031710 162701 004000
9077 031714 162702 000040
9078 031720 010006
9079 031722 012637 000006
9080 031726 012637 000004
9081 031732 010137 031754
9082 031736 010237 031756
9083 031742 012603
9084 031744 012602
9085 031746 012601
9086 031750 012600
9087 031752 000207
9088 031754 000000
9089 031756 000000
9090
9091
9092
9093
9094
9095
9096
9097
9098
9099
9100
9101
9102
9103
9104 031760
9105 031760 104407
9106
9107
9108 031762 021627 001002
9109 031766 101002
9110 031770 000137 033000
9111 031774 032777 040000 147136
9112 032002 001131
9113
9114 032004 000416
9115
9116 032006 013746 000004
9117 032012 012737 032032 000004
9118 032020 005737 177060
9119 032024 012637 000004
9120 032030 000500
9121 032032 022626
9122 032034 012637 000004
9123 032040 000440
9124 032042
9125 032042 032777 000400 147070
9126 032050 001421
9127 032052 005046
9128 032054 117716 147060
9129 032060 001414
9130 032062 022716 000111
9131 032066 002411

```

```

$SCROUT: SUB #4000,R1
$SIZE: SUB #40,R2 ;;DROP BACK
MOV RO,SP ;;RESTORE THE STACK
MOV (SP)+,2#ERRVEC+2 ;;RESTORE ERROR VECTOR
MOV (SP)+,2#ERRVEC
MOV R1,$LSTAD ;;LAST ADDRESS
MOV R2,$LSTBK ;;LAST BANK
MOV (SP)+,R3 ;;RESTORE R3
MOV (SP)+,R2 ;;RESTORE R2
MOV (SP)+,R1 ;;RESTORE R1
MOV (SP)+,RO ;;RESTORE RO
RTS PC
$LSTAD: .WORD 0 ;;CONTAINS THE LAST ADDRESS
$LSTBK: .WORD 0 ;;CONTAINS THE LAST BANK
.SBTTL SCOPE HANDLER ROUTINE

;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1 LOOP ON TEST
;SW11=1 INHIBIT ITERATIONS
;SW09=1 LOOP ON ERROR
;SW08=1 LOOP ON TEST IN SWR<7:0>
;CALL
;* SCOPE ;;SCOPE=IOT

$SCOPE:
;;GO TO ERROR ROUTINE IF RETURN PC LESS THAN 1002
;;OTHERWISE CONTINUE
CMP (SP),#1002 ;;UNEXPECTED TRAP OR INTERRUPT
BHI 1$ ;;ARE TRAPPED HERE VIA IOT
JMP $ERROR ;;GO PROCESS UNEXPECTED TRAP
1$: BIT #BIT14,2$SWR ;;LOOP ON PRESENT TEST?
BNE $OVER ;;YES IF SW14=1
;####START OF CODE FOR THE XOR TESTER####
$XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
;;THIS INSTRUCTION TO A "NOP" (NOP=240)
MOV 2#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
MOV #5$,2#ERRVEC ;;SET FOR TIMEOUT
TST 2#177060 ;;TIME OUT ON XOR?
MOV (SP)+,2#ERRVEC ;;RESTORE THE ERROR VECTOR
BR $SVLAD ;;GO TO THE NEXT TEST
5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
MOV (SP)+,2#ERRVEC ;;RESTORE THE ERROR VECTOR
BR 7$ ;;LOOP ON THE PRESENT TEST
6$;####END OF CODE FOR THE XOR TESTER####
BIT #BIT08,2$SWR ;;LOOP ON SPEC. TEST?
BEQ 2$ ;;BR IF NO
CLR -(SP) ;;CLEAR A TEMP. LOCATION
MOVB 2$SWR,(SP) ;;PICKUP THE DESIRED TEST NUMBER
BEQ 8$ ;;BRANCH IF BAD TEST NUMBER IN SWR
CMP #111,(SP) ;;CHECK THE NUMBER IN THE SWR
BLT 8$ ;;BRANCH IF TEST NUMBER IS OUT OF RANGE

```


9132	032070	011637	001102		MOV	(SP), \$STNM	:: UPDATE THE TEST NUMBER	
9133	032074	005316			DEC	(SP)	:: BACKUP BY ONE	
9134	032076	006316			ASL	(SP)	:: SCALE THE TEST NUMBER AS AN INDEX	
9135	032100	062716	032304		ADD	\$\$SWOBTBL, (SP)	:: FORM THE ADDRESS OF TEST POINTER	
9136	032104	013637	001106		MOV	2(SP)+, \$LPADR	:: SET LOOP ADDRESS TO DESIRED TEST	
9137	032110	000466			BR	\$OVER	:: GO LOOP ON THE TEST	
9138	032112	005726		8\$:	TST	(SP)+	:: CLEAN THE BAD TEST NUMBER OFF OF THE STACK	
9139	032114	105737	001103	2\$:	TSTB	\$ERFLG	:: HAS AN ERROR OCCURRED?	
9140	032120	001421			BEQ	3\$:: BR IF NO	
9141	032122	123737	001115	001103	CMPB	\$ERMAX, \$ERFLG	:: MAX. ERRORS FOR THIS TEST OCCURRED?	
9142	032130	101015			BHI	3\$:: BR IF NO	
9143	032132	032777	001000	147000	BIT	#BIT09, 2SWR	:: LOOP ON ERROR?	
9144	032140	001404			BEQ	4\$:: BR IF NO	
9145	032142	013737	001110	001106	7\$:	MOV	\$LPERR, \$LPADR	:: SET LOOP ADDRESS TO LAST SCOPE
9146	032150	000446			BR	\$OVER		
9147	032152	105037	001103		4\$:	CLRB	\$ERFLG	:: ZERO THE ERROR FLAG
9148	032156	005037	001262		CLR	\$TIMES	:: CLEAR THE NUMBER OF ITERATIONS TO MAKE	
9149	032162	000415			BR	1\$:: ESCAPE TO THE NEXT TEST	
9150	032164	032777	004000	146746	3\$:	BIT	#BIT11, 2SWR	:: INHIBIT ITERATIONS?
9151	032172	001011			BNE	1\$:: BR IF YES	
9152	032174	005737	001304		TST	\$PASS	:: IF FIRST PASS OF PROGRAM	
9153	032200	001406			BEQ	1\$:: INHIBIT ITERATIONS	
9154	032202	005237	001104		INC	\$ICNT	:: INCREMENT ITERATION COUNT	
9155	032206	023737	001262	001104	CMP	\$TIMES, \$ICNT	:: CHECK THE NUMBER OF ITERATIONS MADE	
9156	032214	002024			BGE	\$OVER	:: BR IF MORE ITERATION REQUIRED	
9157	032216	012737	000001	001104	1\$:	MOV	#1, \$ICNT	:: REINITIALIZE THE ITERATION COUNTER
9158	032224	013737	032302	001262	MOV	\$MXCNT, \$TIMES	:: SET NUMBER OF ITERATIONS TO DO	
9159	032232	105237	001102		9\$:	INCB	\$STNM	:: COUNT TEST NUMBERS
9160	032236	113737	001102	001302	MOV	\$STNM, \$STNM	:: SET TEST NUMBER IN APT MAILBOX	
9161	032244	011637	001106		MOV	(SP), \$LPADR	:: SAVE SCOPE LOOP ADDRESS	
9162	032250	011637	001110		MOV	(SP), \$LPERR	:: SAVE ERROR LOOP ADDRESS	
9163	032254	005037	001264		CLR	\$ESCAPE	:: CLEAR THE ESCAPE FROM ERROR ADDRESS	
9164	032260	112737	000001	001115	MOV	#1, \$ERMAX	:: ONLY ALLOW ONE(1) ERROR ON NEXT TEST	
9165	032266	013777	001102	146646	SOVER:	MOV	\$STNM, 2DISPLAY	:: DISPLAY TEST NUMBER
9166	032274	013716	001106		MOV	\$LPADR, (SP)	:: FUDGE RETURN ADDRESS	
9167	032300	000002			RTI		:: FIXES PS	
9168	032302	003720			9\$:	MOV	\$MXCNT, 2000.	:: MAX. NUMBER OF ITERATIONS
9169	032304				9\$:	MOV	\$\$SWOBTBL:	
9170	032304	003120			.WORD	TST1+2	:: STARTING ADDRESS OF TEST 1	
9171	032306	003236			.WORD	TST2+2	:: STARTING ADDRESS OF TEST 2	
9172	032310	003324			.WORD	TST3+2	:: STARTING ADDRESS OF TEST 3	
9173	032312	004350			.WORD	TST4+2	:: STARTING ADDRESS OF TEST 4	
9174	032314	004452			.WORD	TST5+2	:: STARTING ADDRESS OF TEST 5	
9175	032316	004630			.WORD	TST6+2	:: STARTING ADDRESS OF TEST 6	
9176	032320	004756			.WORD	TST7+2	:: STARTING ADDRESS OF TEST 7	
9177	032322	005062			.WORD	TST10+2	:: STARTING ADDRESS OF TEST 10	
9178	032324	006006			.WORD	TST11+2	:: STARTING ADDRESS OF TEST 11	
9179	032326	006124			.WORD	TST12+2	:: STARTING ADDRESS OF TEST 12	
9180	032330	006234			.WORD	TST13+2	:: STARTING ADDRESS OF TEST 13	
9181	032332	006476			.WORD	TST14+2	:: STARTING ADDRESS OF TEST 14	
9182	032334	006630			.WORD	TST15+2	:: STARTING ADDRESS OF TEST 15	
9183	032336	007040			.WORD	TST16+2	:: STARTING ADDRESS OF TEST 16	
9184	032340	007230			.WORD	TST17+2	:: STARTING ADDRESS OF TEST 17	
9185	032342	007352			.WORD	TST20+2	:: STARTING ADDRESS OF TEST 20	
9186	032344	010140			.WORD	TST21+2	:: STARTING ADDRESS OF TEST 21	
9187	032346	010536			.WORD	TST22+2	:: STARTING ADDRESS OF TEST 22	

9188	032350	010750	.WORD	TST23+2	:: STARTING ADDRESS OF TEST	23
9189	032352	011166	.WORD	TST24+2	:: STARTING ADDRESS OF TEST	24
9190	032354	011312	.WORD	TST25+2	:: STARTING ADDRESS OF TEST	25
9191	032356	011444	.WORD	TST26+2	:: STARTING ADDRESS OF TEST	26
9192	032360	011622	.WORD	TST27+2	:: STARTING ADDRESS OF TEST	27
9193	032362	011754	.WORD	TST30+2	:: STARTING ADDRESS OF TEST	30
9194	032364	012250	.WORD	TST31+2	:: STARTING ADDRESS OF TEST	31
9195	032366	012444	.WORD	TST32+2	:: STARTING ADDRESS OF TEST	32
9196	032370	012700	.WORD	TST33+2	:: STARTING ADDRESS OF TEST	33
9197	032372	013066	.WORD	TST34+2	:: STARTING ADDRESS OF TEST	34
9198	032374	013320	.WORD	TST35+2	:: STARTING ADDRESS OF TEST	35
9199	032376	013506	.WORD	TST36+2	:: STARTING ADDRESS OF TEST	36
9200	032400	013730	.WORD	TST37+2	:: STARTING ADDRESS OF TEST	37
9201	032402	014116	.WORD	TST40+2	:: STARTING ADDRESS OF TEST	40
9202	032404	014310	.WORD	TST41+2	:: STARTING ADDRESS OF TEST	41
9203	032406	014504	.WORD	TST42+2	:: STARTING ADDRESS OF TEST	42
9204	032410	014710	.WORD	TST43+2	:: STARTING ADDRESS OF TEST	43
9205	032412	015114	.WORD	TST44+2	:: STARTING ADDRESS OF TEST	44
9206	032414	015574	.WORD	TST45+2	:: STARTING ADDRESS OF TEST	45
9207	032416	016000	.WORD	TST46+2	:: STARTING ADDRESS OF TEST	46
9208	032420	016204	.WORD	TST47+2	:: STARTING ADDRESS OF TEST	47
9209	032422	016422	.WORD	TST50+2	:: STARTING ADDRESS OF TEST	50
9210	032424	016626	.WORD	TST51+2	:: STARTING ADDRESS OF TEST	51
9211	032426	017032	.WORD	TST52+2	:: STARTING ADDRESS OF TEST	52
9212	032430	017156	.WORD	TST53+2	:: STARTING ADDRESS OF TEST	53
9213	032432	017302	.WORD	TST54+2	:: STARTING ADDRESS OF TEST	54
9214	032434	017426	.WORD	TST55+2	:: STARTING ADDRESS OF TEST	55
9215	032436	017552	.WORD	TST56+2	:: STARTING ADDRESS OF TEST	56
9216	032440	017676	.WORD	TST57+2	:: STARTING ADDRESS OF TEST	57
9217	032442	017742	.WORD	TST60+2	:: STARTING ADDRESS OF TEST	60
9218	032444	020014	.WORD	TST61+2	:: STARTING ADDRESS OF TEST	61
9219	032446	020272	.WORD	TST62+2	:: STARTING ADDRESS OF TEST	62
9220	032450	020454	.WORD	TST63+2	:: STARTING ADDRESS OF TEST	63
9221	032452	020712	.WORD	TST64+2	:: STARTING ADDRESS OF TEST	64
9222	032454	021100	.WORD	TST65+2	:: STARTING ADDRESS OF TEST	65
9223	032456	021604	.WORD	TST66+2	:: STARTING ADDRESS OF TEST	66
9224	032460	022014	.WORD	TST67+2	:: STARTING ADDRESS OF TEST	67
9225	032462	022120	.WORD	TST70+2	:: STARTING ADDRESS OF TEST	70
9226	032464	022540	.WORD	TST71+2	:: STARTING ADDRESS OF TEST	71
9227	032466	023160	.WORD	TST72+2	:: STARTING ADDRESS OF TEST	72
9228	032470	023270	.WORD	TST73+2	:: STARTING ADDRESS OF TEST	73
9229	032472	024122	.WORD	TST74+2	:: STARTING ADDRESS OF TEST	74
9230	032474	024600	.WORD	TST75+2	:: STARTING ADDRESS OF TEST	75
9231	032476	025014	.WORD	TST76+2	:: STARTING ADDRESS OF TEST	76
9232	032500	025230	.WORD	TST77+2	:: STARTING ADDRESS OF TEST	77
9233	032502	025310	.WORD	TST100+2	:: STARTING ADDRESS OF TEST	100
9234	032504	025502	.WORD	TST101+2	:: STARTING ADDRESS OF TEST	101
9235	032506	025726	.WORD	TST102+2	:: STARTING ADDRESS OF TEST	102
9236	032510	026054	.WORD	TST103+2	:: STARTING ADDRESS OF TEST	103
9237	032512	026400	.WORD	TST104+2	:: STARTING ADDRESS OF TEST	104
9238	032514	027132	.WORD	TST105+2	:: STARTING ADDRESS OF TEST	105
9239	032516	027334	.WORD	TST106+2	:: STARTING ADDRESS OF TEST	106
9240	032520	027554	.WORD	TST107+2	:: STARTING ADDRESS OF TEST	107
9241	032522	027770	.WORD	TST110+2	:: STARTING ADDRESS OF TEST	110
9242	032524	030466	.WORD	TST111+2	:: STARTING ADDRESS OF TEST	111
9243	032526	031244	.WORD	SEOP+2	:: ADDRESS OF END OF PASS	


```

9244 032530 042710          .WORD  ABTFAIL+2          ;ADDRESS OF ABORT FAILURE HANDLER
9245
9246          .SBTTL  APT COMMUNICATIONS ROUTINE
9247
9248          ;*****
9249 032532 112737 000001 032776 $ATY1:  MOVB  #1,$FFLG          ;;TO REPORT FATAL ERROR
9250 032540 112737 000001 032774 $ATY3:  MOVB  #1,$MFLG          ;;TO TYPE A MESSAGE
9251 032546 000403          BR      $ATYC
9252 032550 112737 000001 032776 $ATY4:  MOVB  #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
9253 032556          $ATYC:
9254 032556 010046          MOV      RO,-(SP)          ;;PUSH RO ON STACK
9255 032560 010146          MOV      R1,-(SP)          ;;PUSH R1 ON STACK
9256 032562 105737 032774          TSTB   $MFLG          ;;SHOULD TYPE A MESSAGE?
9257 032566 001450          BEQ     5$              ;;IF NOT: BR
9258 032570 122737 000001 001316          CMPB   #APTENV,$ENV      ;;OPERATING UNDER APT?
9259 032576 001031          BNE     3$              ;;IF NOT: BR
9260 032600 132737 000100 001317          BITB   #APTPOOL,$ENVM   ;;SHOULD SPOOL MESSAGES?
9261 032606 001425          BEQ     3$              ;;IF NOT: BR
9262 032610 017600 000004          MOV     @4(SP),RO        ;;GET MESSAGE ADDR.
9263 032614 062766 000002 000004          ADD     #2,4(SP)         ;;BUMP RETURN ADDR.
9264 032622 005737 001276          1$:    TST     $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
9265 032626 001375          BNE     1$              ;;IF NOT: WAIT
9266 032630 010037 001312          MOV     RO,$MSGAD       ;;PUT ADDR IN MAILBOX
9267 032634 105720          2$:    TSTB   (RO)+         ;;FIND END OF MESSAGE
9268 032636 001376          BNE     2$
9269 032640 163700 001312          SUB     $MSGAD,RO        ;;SUB START OF MESSAGE
9270 032644 006200          ASR     RO              ;;GET MESSAGE LNTH IN WORDS
9271 032646 010037 001314          MOV     RO,$MSGGLT      ;;PUT LENGTH IN MAILBOX
9272 032652 012737 000004 001276          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
9273 032660 000413          BR      5$
9274 032662 017637 000004 032706 3$:    MOV     @4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
9275 032670 062766 000002 000004          ADD     #2,4(SP)         ;;BUMP RETURN ADDRESS
9276 032676 013746 177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
9277 032702 004737 042720          JSR     PC,$TYPE        ;;CALL TYPE MACRO
9278 032706 000000          4$:    .WORD  0
9279 032710          5$:
9280 032710 105737 032776          10$:   TSTB   $FFLG          ;;SHOULD REPORT FATAL ERROR?
9281 032714 001416          BEQ     12$             ;;IF NOT: BR
9282 032716 005737 001316          TST     $ENV            ;;RUNNING UNDER APT?
9283 032722 001413          BEQ     12$             ;;IF NOT: BR
9284 032724 005737 001276          11$:   TST     $MSGTYPE      ;;FINISHED LAST MESSAGE?
9285 032730 001375          BNE     11$             ;;IF NOT: WAIT
9286 032732 017637 000004 001300          MOV     @4(SP),$FATAL   ;;GET ERROR #
9287 032740 062766 000002 000004          ADD     #2,4(SP)         ;;BUMP RETURN ADDR.
9288 032746 005237 001276          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
9289 032752 105037 032776          12$:   CLRB   $FFLG          ;;CLEAR FATAL FLAG
9290 032756 105037 032775          CLRB   $LFLG           ;;CLEAR LOG FLAG
9291 032762 105037 032774          CLRB   $MFLG           ;;CLEAR MESSAGE FLAG
9292 032766 012601          MOV     (SP)+,R1        ;;POP STACK INTO R1
9293 032770 012600          MOV     (SP)+,RO        ;;POP STACK INTO RO
9294 032772 000207          RTS     PC              ;;RETURN
9295 032774          000          $MFLG: .BYTE  0          ;;MESSG. FLAG
9296 032775          000          $LFLG: .BYTE  0          ;;LOG FLAG
9297 032776          000          $FFLG: .BYTE  0          ;;FATAL FLAG
9298          033000          .EVEN
9299          000200          APTSIZE=200

```



```

9300      000001
9301      000100
9302      000040
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317 033000
9318 033000 104407
9319 033002 105237 001103
9320 033006 001775
9321 033010 013777 001102 146124
9322 033016 032777 002000 146114
9323 033024 001402
9324 033026 104401 001266
9325 033032 005237 001112
9326 033036 011637 001116
9327 033042 162737 000002 001116
9328 033050 117737 146042 001114
9329 033056 032777 020000 146054
9330 033064 001055
9331 033066 021627 001002
9332 033072 101046
9333
9334 033074 016637 000004 001116
9335 033102 162737 000002 001116
9336 033110 104401 033154
9337 033114 013746 001116
9338 033120 104402
9339 033122 104401 033162
9340 033126 162716 000004
9341 033132 011637 001116
9342 033136 013746 001116
9343 033142 104402
9344 033144 104401 001273
9345 033150 022626
9346 033152 000422
9347 033154 050200 036503 000040
9348 033162 020040 047125 054105
9349 033170 042520 052103 042105
9350 033176 052040 040522 020120
9351 033204 047524 000040
9352
9353 033210
9354 033210 004737 033322
9355 033214 104401 001273

```

```

APTENV=001
APTSPool=100
APTCSUP=040
.SBTTL ERROR HANDLER ROUTINE

*****
*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
*AND GO TO TYPERR ON ERROR
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW15=1 HALT ON ERROR
*SW13=1 INHIBIT ERROR TYPEOUTS
*SW10=1 BELL ON ERROR
*SW09=1 LOOP ON ERROR
*CALL
*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER

$ERROR:
7$:      CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
          INCB      $ERFLG      ;;SET THE ERROR FLAG
          BEQ      7$      ;;DON'T LET THE FLAG GO TO ZERO
          MOV      $TSTNM,$DISPLAY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
          BIT      #BIT10,$SWR      ;;BELL ON ERROR?
          BEQ      1$      ;;NO - SKIP
          TYPE     $BELL      ;;RING BELL
          INC      $ERTTL      ;;COUNT THE NUMBER OF ERRORS
          MOV      (SP),$ERRPC      ;;GET ADDRESS OF ERROR INSTRUCTION
          SUB      #2,$ERRPC
          MOV      $ERRPC,$ITEMB      ;;STRIP AND SAVE THE ERROR ITEM CODE
          BIT      #BIT13,$SWR      ;;SKIP TYPEOUT IF SET
          BNE      20$      ;;SKIP TYPEOUTS
          CMP      (SP),#1002      ;;IF RETURN PC LESS THAN 1002
          BHI      12$      ;;ERROR IS ILLEGAL TRAP
          ;;PROCESS UNEXPECTED TRAP OR INTERRUPT
          MOV      4(SP),$ERRPC      ;;GET PC AT TIME OF FALSE TRAP
          SUB      #2,$ERRPC      ;;ADJUST PC
          TYPE     10$      ;;TYPE HEADER
          MOV      $ERRPC,-(SP)      ;;SAVE $ERRPC FOR TYPEOUT
          TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
          TYPE     11$
          SUB      #4,(SP)      ;;GET FALSE TRAP VECTOR ADDR
          MOV      (SP),$ERRPC
          MOV      $ERRPC,-(SP)      ;;SAVE $ERRPC FOR TYPEOUT
          TYP0C      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
          TYPE     $CRLF
          CMP      (SP)+,(SP)+      ;;POP FALSE TRAP VECTOR PC&ADDR
          BR       20$
10$:     .ASCIZ    '<200>'PC= '
11$:     .ASCIZ    ' UNEXPECTED TRAP TO '

          .EVEN
12$:     JSR      PC,TYPERR      ;;GO TO USER ERROR ROUTINE
          TYPE     $CRLF

```



```

9356 033220                20$:
9357 033220 122737 000001 001316      CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
9358 033226 001007                BNE     2$              ;;NO SKIP APT ERROR REPORT
9359 033230 113737 001114 033242      MOVB   $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
9360 033236 004737 032550                JSR    PC,$ATY4        ;;REPORT FATAL ERROR TO APT
9361 033242 000                21$: .BYTE   0
9362 033243 000                .BYTE   0
9363 033244 000777                22$: BR     22$              ;; APT ERROR LOOP
9364 033246 005777 145666                2$:  TST   $SWR          ;; HALT ON ERROR
9365 033252 100002                BPL    3$              ;; SKIP IF CONTINUE
9366 033254 000000                HALT                   ;; HALT ON ERROR!
9367 033256 104407                CKSWR                   ;; TEST FOR CHANGE IN SOFT-SWR
9368 033260 032777 001000 145652      3$:  BIT   #BIT09,$SWR   ;; LOOP ON ERROR SWITCH SET?
9369 033266 001402                BEQ    4$              ;; BR IF NO
9370 033270 013716 001110                MOV   $LPERR,(SP)      ;; FUDGE RETURN FOR LOOPING
9371 033274 005737 001264                4$:  TST   $ESCAPE       ;; CHECK FOR AN ESCAPE ADDRESS
9372 033300 001402                BEQ    5$              ;; BR IF NONE
9373 033302 013716 001264                MOV   $ESCAPE,(SP)    ;; FUDGE RETURN ADDRESS FOR ESCAPE
9374 033306
9375 033306 022737 031432 000042        5$:  CMP   #SENDAD,$#42  ;; ACT-11 AUTO-ACCEPT?
9376 033314 001001                BNE    6$              ;; BRANCH IF NO
9377 033316 000000                HALT                   ;; YES
9378 033320
9379 033320 000002                6$:  RTI                    ;; RETURN
9380
9381 ;*****
9382 ;SBTTL TYPE ERROR ROUTINE
9383 ;*ENTRY JSR PC,TYPERR
9384 ;*RETURN RTS PC
9385 ;*
9386 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9387 ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
9388 ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
9389 ;*THE ERROR.
9390 ;*****
9390 033322 104413      TYPERR: SAVREG
9391 033324 113737 001102 001302      MOVB   $STNM,$STNM    ;GET TEST NUMBER FOR REPORT
9392 033332 042737 177400 001302      BIC   #177400,$STNM  ;CLEAR UNUSED BITS
9393 033340 113700 001114                MOVB   $ITEMB,R0     ;ENTER ERROR NUMBER
9394 033344 042700 177400                BIC   #177400,R0     ;CLEAR UNUSED BITS
9395 033350 005300                DEC   R0              ;FORM INDEX FOR ERROR TABLE
9396 033352 006300                ASL   R0
9397 033354 006300                ASL   R0
9398 033356 006300                ASL   R0
9399 033360 062700 001360                1$:  ADD   #ERRTB,R0     ;FORM ADDRESS OF ERROR ENTRY
9400 033364 012037 033400                MOV   (R0)+,2$       ;GET EM POINTER
9401 033370 001404                BEQ   3$              ;BRANCH IF THERE ISN'T ONE
9402 033372 104401 001273                TYPE  , $CRLF        ;TYPE CARRIAGE RETURN LINE FEED
9403 033376 104401                TYPE  ;              ;TYPE ERROR MESSAGE (EM)
9404 033400 000000                2$:  .WORD  0           ;EM POINTER GOES HERE
9405 033402 012037 033416                3$:  MOV   (R0)+,4$     ;GET DH POINTER
9406 033406 001404                BEQ   5$              ;BRANCH IF THERE ISN'T ONE
9407 033410 104401 001273                TYPE  , $CRLF        ;TYPE CR-LF
9408 033414 104401                TYPE  ;              ;TYPE DATA HEADER
9409 033416 000000                4$:  .WORD  0           ;DH POINTER GOES HERE
9410 033420 012001                5$:  MOV   (R0)+,R1     ;GET DT POINTER
9411 033422 001445                BEQ   20$            ;BRANCH IF THERE ARE NONE

```


9412	033424	005004		CLR	R4		;SET INDENT SWITCH
9413	033426	012000		MOV	(R0)+,R0		;GET DF POINTER
9414	033430	012002		MOV	(R0)+,R2		;STORE NUMBER OF DH'S
9415	033432	104401	001273	TYPE	,SCRLF		
9416	033436	112003		10\$:	MOVB	(R0)+,R3	;GET & STORE NUMBER OF DATA WORDS
9417	033440	105720			TSTB	(R0)+	;BUMP PAST FORMAT WORD
9418	033442	005703			TST	R3	;TEST IF ANY DATA FOR THIS HEADER
9419	033444	001416			BEQ	14\$;NO - SKIP DATA PRINT
9420	033446	005704			TST	R4	;CHECK FOR INDENT
9421	033450	001004			BNE	12\$;YES, GO INDENT
9422	033452	013146		11\$:	MOV	2(R1)+,-(SP)	;PUT FIRST DATA WORD ON STACK
9423	033454	104402			TYPOC		;TYPE IT
9424	033456	005303			DEC	R3	;MORE DATA WORDS
9425	033460	001403			BEQ	13\$;NO-BRANCH
9426	033462	104401	047501	12\$:	TYPE	,SPACE2	;TYPE SEPARATORS
9427	033466	000771			BR	11\$;LOOP
9428	033470	104401	001273	13\$:	TYPE	,SCRLF	;TYPE <CR><LF>
9429	033474	005710			TST	(R0)	;CHECK IF NEXT HEADER AVAILABLE
9430	033476	001401			BEQ	14\$;NO, DO NOT CHANGE INDENT
9431	033500	005104			COM	R4	;CHANGE INDENT
9432	033502	005302		14\$:	DEC	R2	;MORE DH'S?
9433	033504	003414			BLE	20\$;NO-BRANCH
9434	033506	012037	033526	15\$:	MOV	(R0)+,18\$;GET NEXT DH POINTER
9435	033512	001751			BEQ	10\$;IF 0 GET DATA
9436	033514	005704			TST	R4	;INDENT?
9437	033516	001402			BEQ	17\$;NO, BRANCH
9438	033520	104401	047501		TYPE	,SPACE2	;TYPE INDENT
9439	033524	104401		17\$:	TYPE		;TYPE DH
9440	033526	000000		18\$:	.WORD	0	;DH POINTER GOES HERE
9441	033530	104401	001273		TYPE	,SCRLF	
9442	033534	000740			BR	10\$;GET DATA
9443	033536	104414		20\$:	RESREG		
9444	033540	005237	001632		INC	ERRCNT	;INCREMENT THE ERROR COUNT
9445	033544	032777	010000 145366		BIT	#SW12,2SWR	;CHECK IF SWITCH 12 SET
9446	033552	001421			BEQ	25\$;NO, RETURN
9447	033554	022737	000024 001632		CMP	#20.,ERRCNT	;CHECK IF ERROR THRESHOLD EXCEEDED
9448	033562	103015			BHIS	25\$;NO, RETURN
9449	033564	104401	052130		TYPE	,ABORT	;TYPE "PROGRAM ABORTED BECAUSE
9450							;ERROR THRESHOLD EXCEEDED"
9451	033570	005737	000042		TST	42	;CHECK IF CHAIN MODE
9452	033574	001407			BEQ	22\$;NO, HALT PROCESSOR
9453	033576	012706	001100		MOV	#STACK,SP	;INITIALIZE STACK
9454	033602	012737	000001 031270		MOV	#1,\$EOPCT	;FORCE END OF PROGRAM
9455	033610	000137	031242		JMP	\$EOP	
9456	033614	000000		22\$:	HALT		
9457	033616	000207		25\$:	RTS	PC	
9458				.SBTTL	NON-EXISTANT MEMORY AND INTERRUPT CHECK HANDLER		
9459				;*:	THIS ROUTINE SETS THE INTERRUPT FLAG AND DOES AN RTI.		
9460				;*:	THIS IS THE INDICATION TO THE ROUTINE CHECKING		
9461				;*:	NON-EXISTANT MEMORY THAT AN INTERRUPT DID OCCUR.		
9462							
9463	033620	005237	001662	NEXINT:	INC	INTSET	;BUMP THE INTERRUPT COUNTER
9464	033624	000002			RTI		
9465				.SBTTL	RK611 INTERRUPT HANDLER		
9466				;*:	MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE. ACTUAL		
9467							


```

9468          ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
9469          ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
9470          ;*      INTERRUPT OCCURRED.
9471
9472 033626 005237 001662      INTHLR: INC      INTSET      ;BUMP THE INTERRUPT FLAG
9473 033632 000002              RTI              ;RETURN.
9474
9475          .SBTTL  MEMORY PARITY ERROR TRAP HANDLER
9476          ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE.  THE REPORT
9477          ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
9478
9479 033634 032777 020000 145276 PERHLR: BIT      #BIT13,JSWR      ;TEST IF INHIBIT REPORT
9480 033642 001003              BNE      1$              ;YES - SKIP
9481 033644 104401 052357      TYPE      ,EM3          ;TYPE PARITY ERROR MESSAGE
9482 033650 104402              TYPOC           ;AND PC VALUE
9483 033652 012737 000001 031270 1$:  MOV      #1,$EOPCT      ;FORCE END OF PROGRAM
9484 033660 012706 001100      MOV      #STACK,SP      ;CLEAN OFF STACK
9485 033664 000137 031242      JMP      $EOP
9486
9487          .SBTTL  LINE CLOCK INTERRUPT HANDLER
9488          ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
9489          ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
9490
9491 033670 005237 001660      LCKHLR: INC      LCLKTK      ;INCREMENT CLOCK TICK COUNTER
9492 033674 042777 000200 145774      BIC      #BIT7,$KWLADD    ;CLEAR MONITOR BIT
9493 033702 000002              RTI
9494
9495          ;*****
9496          .SBTTL  OPTION PRESENT TEST AND SETUP
9497          ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
9498          ;*      LINE CLOCK ARE ON THE SYSTEM.  FLAGS ARE SET IF PRESENT; CLEARED
9499          ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
9500 033704 104413              OPTTST: SAVREG
9501 033706 012701 000004      MOV      #4,R1          ;SET POINTER TO NEM VECTOR ADDRESS
9502 033712 012146              MOV      (R1)+,-(SP)      ;STORE VECTOR CONTENTS
9503 033714 011146              MOV      (R1),-(SP)
9504 033716 012711 000340      MOV      #PR7,(R1)       ;SET PRIORITY
9505 033722 012741 033620      MOV      #NEXINT,-(R1)   ;SET VECTOR TO NEM TEST HANDLER
9506 033726 005037 001662      CLR      INTSET         ;CLEAR INTERRUPT COUNTER
9507 033732 012777 000001 145742      MOV      #1,$MMCSR1      ;LOAD CSR FOR BANK 2
9508 033740 000240              NOP
9509 033742 000240              NOP
9510 033744 000240              NOP
9511 033746 005737 001662      TST      INTSET         ;CHECK IF INTERRUPT OCCURRED
9512          ;*      BECAUSE OF NEM ON PARITY OPTION REFER.
9513 033752 001011              BNE      2$              ;YES - SKIP PARITY SETUP
9514 033754 052737 000200 001656      BIS      #PARBK0,OPTFLG  ;SET PARITY OPTION FLAG
9515 033762 013700 001706      MOV      MMVECA,R0       ;SET POINTER TO VECTOR
9516 033766 012720 033634      MOV      #PERHLR,(R0)+   ;INSERT HANDLER ADDRESS
9517 033772 012710 000340      MOV      #PR7,(R0)       ;INSERT PRIORITY
9518 033776 005037 001662      CLR      INTSET         ;CLEAR INTERRUPT FLAG
9519 034002 012777 000001 145674      MOV      #1,$MMCSR2      ;LOAD CSR FOR BANK 1
9520 034010 000240              NOP
9521 034012 000240              NOP
9522 034014 000240              NOP
9523 034016 005737 001662      TST      INTSET         ;TEST IN INTERRUPT OCCURRED

```



```

9524 034022 001003          BNE      3$          ;YES - SKIP SETUP
9525 034024 052737 000100 001656 3$:  BIS      #PARBK1,OPTFLG ;SET FLAG
9526 034032 005037 001662          CLR      INTSET      ;CLEAR INTERRUPT COUNTER
9527 034036 013700 001700          MOV      KWLVEC,RO    ;SET POINTER TO VECTOR
9528 034042 012720 033670          MOV      #LCKHLR,(RO)+ ;INSERT ADDRESS OF INTERRUPT HDLR
9529 034046 012710 000340          MOV      #PR7,(RO)    ;INSERT PRIORITY
9530 034052 012777 000100 145616  MOV      #BIT6,@KWLADD ;LOAD KW11-L FOR INTERRUPT ENABLE
9531 034060 000240          NOP
9532 034062 000240          NOP
9533 034064 000240          NOP
9534 034066 005737 001662          TST      INTSET      ;TEST IN NEM ON KW11-P REFERENCE
9535 034072 001003          BNE      4$          ;THIS BRANCH WILL BYPASS SET UP OF
9536                                     ;CLOCK OPTION
9537 034074 052737 100000 001656 4$:  BIS      #LCLKPR,OPTFLG ;SET CLOCK PRESENT FLAG
9538 034102 012701 000006          MOV      #6,R1       ;RESTORE OLD VECTOR
9539 034106 005037 001662          CLR      INTSET      ;CLEAR INT FLAG
9540 034112 012611          MOV      (SP)+,(R1)
9541 034114 012641          MOV      (SP)+,-(R1)
9542 034116 104414          RESREG
9543 034120 000207          RTS      PC

```

```

9544
9545 ;*****
9546 .SBTTL LOOP ON INTERNAL ERROR
9547 ;*
9548 ;* THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
9549 ;* IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
9550 ;* TO BE EXECUTED ON ERROR.

```

```

9551 034122 032777 001000 145010 SCOP1$: BIT      #SW9,@SWR      ;CHECK IF LOOP ON ERROR
9552 034130 001405          BEQ      5$          ;NO, CONTINUE
9553 034132 105737 001103          TSTB     $ERFLG      ;CHECK IF ERROR OCCURRED
9554 034136 001402          BEQ      5$
9555 034140 013716 001110          MOV      $LPERR,(SP) ;LOAD ERROR RETURN
9556 034144 000002          RTI              ;RETURN

```

```

9557 .SBTTL LINE CLOCK CALIBRATE
9558 ;*
9559 ;* WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS
9560 ;* TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
9561 ;* THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
9562 ;* LINE CLOCK SIMULATOR.

```

```

9563 034146 005037 001660          CLKCAL: CLR      LCLKTK      ;CLEAR TICK COUNTER
9564 034152 032737 100000 001656  BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
9565 034160 001004          BNE      1$          ;YES - SKIP
9566 034162 012737 000020 001654  MOV      #16.,TIMCNT ;ELSE PRESET TIMCNT
9567 034170 000410          BR       2$          ;AND EXIT
9568 034172 005737 001662          1$:  TST      INTSET      ;TEST IF INTERRUPT HAS OCCURRED
9569 034176 001005          BNE      2$          ;YES- ABORT CALIBRATION
9570 034200 005737 001660          TST      LCLKTK      ;WAIT FOR CLOCK TICK
9571 034204 001772          BEQ      1$          ;NOT YET - LOOP
9572 034206 005037 001660          CLR      LCLKTK      ;CLEAR TICK COUNT
9573 034212 000207          2$:  RTS      PC          ;RETURN

```

```

9574 .SBTTL LINE CLOCK SIMULATION ROUTINE
9575 ;*
9576 ;* THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO
9577 ;* DO THIS THE VALUE STORED IN MILCNT IS USED AS THE
9578 ;* BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT
9579 ;* AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE

```



```

9580      ;*      TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE
9581      ;*      LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT
9582      ;*      IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
9583      ;*      TICK).
9584      ;*
9585      ;*      THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND
9586      ;*      AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.
9587      ;*
9588
9589      034214 010046      MYTIME: MOV      RO, -(SP)      ;SAVE RO
9590      034216 013700 001652      MOV      MLCNT, RO      ;SET COUNT
9591      034222 005737 001662      1$:     TST      INTSET      ;TEST IF INTERRUPT SET
9592      034226 001012      BNE      2$      ;YES - SKIP
9593      034230 005300      DEC      RO      ;DECREMENT COUNT TO ZERO
9594      034232 001373      BNE      1$
9595      034234 005337 001654      DEC      TIMCNT      ;DECREMENT TIMCNT
9596      034240 001005      BNE      2$      ;IF NOT ZERO - EXIT
9597      034242 005237 001660      INC      LCLKTK      ;ELSE BUMP TICK COUNTER
9598      034246 012737 000020 001654      MOV      #16, TIMCNT ;RESET TIME COUNT
9599      034254 012600      MOV      (SP)+, RO      ;RESTORE RO
9600      034256 000207      RTS      PC      ;RETURN
9601
9602      .SBTTL  WAIT FOR INTERRUPT ROUTINE
9603      ;*      THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL
9604      ;*      SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE
9605      ;*      WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR
9606      ;*      IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED
9607      ;*      (BUT NOT PRINTED IN THE ROUTINE) AND THEN RETURNS TO THE
9608      ;*      LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE
9609      ;*      RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL
9610      ;*      BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.
9611
9612      034260 104413      IWAT8S: SAVREG      ;ENTRY FOR 8 SECOND WAIT
9613      034262 012700 000764      MOV      #500, RO
9614      034266 000463      BR      WATSRT
9615      034270 104413      IWAT1M: SAVREG      ;ENTRY FOR 1 MIN WAIT
9616      034272 012700 007246      MOV      #3750, RO
9617      034276 000457      BR      WATSRT
9618      034300 104413      IWAT2S: SAVREG      ;ENTRY FOR 2 SECOND WAIT
9619      034302 012700 000200      MOV      #128, RO
9620      034306 000453      BR      WATSRT
9621      034310 104413      IWAT1S: SAVREG      ;ENTRY FOR 1 SECOND WAIT
9622      034312 012700 000077      MOV      #63, RO
9623      034316 000447      BR      WATSRT
9624      034320 104413      IWAT159: SAVREG      ;ENTRY FOR 160 MS WAIT
9625      034322 012700 000012      MOV      #10, RO
9626      034326 000443      BR      WATSRT
9627      034330 104413      IWAT144: SAVREG      ;ENTRY FOR 144 MS WAIT
9628      034332 012700 000011      MOV      #9, RO
9629      034336 000437      BR      WATSRT
9630      034340 104413      IWAT128: SAVREG      ;ENTRY FOR 128 MS WAIT
9631      034342 012700 000010      MOV      #8, RO
9632      034346 000433      BR      WATSRT
9633      034350 104413      IWAT112: SAVREG      ;ENTRY FOR 112 MS WAIT
9634      034352 012700 000007      MOV      #7, RO
9635      034356 000427      BR      WATSRT

```


9636	034360	104413			IWAT96:	SAVREG			;ENTRY FOR 96 MS WAIT
9637	034362	012700	000006			MOV	#6,RO		
9638	034366	000423				BR	WATSRT		
9639	034370	104413			IWAT80:	SAVREG			;ENTRY FOR 80 MS WAIT
9640	034372	012700	000005			MOV	#5,RO		
9641	034376	000417				BR	WATSRT		
9642	034400	104413			IWAT64:	SAVREG			;ENTRY FOR 64 MS WAIT
9643	034402	012700	000004			MOV	#4,RO		
9644	034406	000413				BR	WATSRT		
9645	034410	104413			IWAT48:	SAVREG			;ENTRY FOR 48 MS WAIT
9646	034412	012700	000003			MOV	#3,RO		
9647	034416	000407				BR	WATSRT		
9648	034420	104413			IWAT32:	SAVREG			;ENTRY FOR 32 MS WAIT
9649	034422	012700	000002			MOV	#2,RO		
9650	034426	000403				BR	WATSRT		
9651	034430	104413			IWAT16:	SAVREG			;ENTRY FOR 16 MS WAIT
9652	034432	012700	000001			MOV	#1,RO		
9653	034436	012746	000000		WATSRT:	MOV	#PRO,-(SP)		;LOAD PRIORITY 0 ON STACK
9654	034442	012746	034450			MOV	#5\$,-(SP)		;LOAD ADDRESS
9655	034446	000002				RTI			
9656									
9657	034450	012737	000020	001654	5\$:	MOV	#16,,TIMCNT		;PRESET TIME COUNTER
9658	034456	004737	034146			JSR	PC,CLKCAL		;GO CALIBRATE THE CLOCK
9659	034462	005737	001662		1\$:	TST	INTSET		;TEST IF INTERRUPT OCCURRED
9660	034466	001036				BNE	3\$;YES - EXIT
9661	034470	032737	100000	001656		BIT	#LCLKPR,OPTFLG		;TEST IF KW11-L AVAILABLE
9662	034476	001002				BNE	2\$;YES - SKIP
9663	034500	004737	034214			JSR	PC,MYTIME		;ELSE CALL SIMULATOR
9664	034504	023700	001660		2\$:	CMP	LCLKTK,RO		;TEST IF ENOUGH TICKS COUNTED
9665	034510	103764				BLO	1\$;NO - LOOP
9666	034512	104420				TGETRK			;ELSE GET '611 REGS
9667	034514	013701	001540			MOV	T.CS1,R1		;PUT CS1 IN R1- STRIP ALL BUT
9668	034520	042701	177741			BIC	#177741,R1		;COMMAND CODE; INDEX INTO TABLE
9669	034524	016137	046636	001372		MOV	CMNDLB(R1),DH2N		;AND SELECT HEADER TO IDENTIFY OPERATION
9670	034532	012737	052427	001370		MOV	#EM4,EM2N		;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
9671	034540	013700	001302			MOV	\$TESTN,RO		;GET NUMBER OF PRESENT TEST
9672	034544	006300				ASL	RO		;SHIFT FOR INDEX
9673	034546	016037	032304	001264		MOV	\$\$SW08TB(RO),\$ESCAPE		;LOAD ESCAPE TO ABORT TESTS
9674	034554	162737	000002	001264		SUB	#2,\$ESCAPE		;BUT GO TO NEXT SCOPE CALL
9675	034562	000402				BR	4\$		
9676	034564	062716	000002		3\$:	ADD	#2,(SP)		;BUMP RETURN AROUND ERROR
9677	034570	104414			4\$:	RESREG			;RESTORE REGS
9678	034572	000002				RTI			;RETURN
9679									
9680					.SBTTL	"L" REGISTER LOADING ROUTINE			
9681					;	THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO			
9682					;	THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT			
9683					;	LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED			
9684					;	FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.			
9685					;				
9686					;	CALL: JSR R4,LRLoad			
9687					;	COMMAND			
9688					;	WORD COUNT			
9689					;	BUS ADDRESS			
9690					;	.BYTE	SECTOR ADDRESS		
9691					;	.BYTE	TRACK ADDRESS		


```

9692          ;*          CYLINDER ADDRESS
9693
9694 034574    LRLoad:
9695 034574 010046    MOV    R0,-(SP)          ;;PUSH R0 ON STACK
9696 034576 010146    MOV    R1,-(SP)          ;;PUSH R1 ON STACK
9697 034600 012701 001600    MOV    #L.CS1,R1        ;;GET ADDRESS OF L REGS
9698 034604 012700 000004    MOV    #4,R0            ;;PRESET COUNT
9699 034610 012421    1$:  MOV    (R4)+,(R1)+    ;;MOVE FIRST FOUR WORDS INTO "L" REGS
9700 034612 005300    DEC    R0                ;;CS1, WC, BA, DA
9701 034614 001375    BNE   1$
9702 034616 013721 001626    MOV    DRVNUM,(R1)+    ;;LOAD DRIVE NUMBER
9703 034622 005721    TST   (R1)+              ;;BUMP PAST ASOF
9704 034624 012411    MOV    (R4)+,(R1)        ;;LOAD DCYL
9705 034626 012601    MOV    (SP)+,R1          ;;POP STACK INTO R1
9706 034630 012600    MOV    (SP)+,R0          ;;POP STACK INTO R0
9707 034632 000204    RTS    R4
9708
9709          .SBTTL  LOAD RK611 FOR OPERATION
9710          ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
9711          ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
9712          ;*      OF THE REGISTER CONTENTS.  L.CS1 IS TRANSFERRED LAST AS IT
9713          ;*      SHOULD BE IF THE GO BIT IS SET.
9714 034634 005037 001662    LOADRK: CLR  INTSET          ;;CLEAR INTERRUPT FLAG.
9715 034640 010046    MOV    R0,-(SP)          ;;STORE REGISTER
9716 034642 010146    MOV    R1,-(SP)
9717 034644 012700 001602    MOV    #L.WC,R0          ;;GET ADDRESS OF SETUP STORAGE WC
9718 034650 010201    MOV    R2,R1              ;;GET BASE ADDRESS
9719 034652 062701 000002    ADD    #2,R1              ;;PUT R1 PAST RKCS1
9720 034656 012021    MOV    (R0)+,(R1)+      ;;LOAD WORD COUNT
9721 034660 012021    MOV    (R0)+,(R1)+      ;;LOAD BUS ADDRESS
9722 034662 012021    MOV    (R0)+,(R1)+      ;;LOAD DISK ADDRESS
9723 034664 012011    MOV    (R0)+,(R1)        ;;LOAD CS2
9724 034666 062701 000006    ADD    #6,R1              ;;BUMP R1 TO ASOF
9725 034672 012021    MOV    (R0)+,(R1)+      ;;LOAD OFFSET
9726 034674 012021    MOV    (R0)+,(R1)+      ;;LOAD CYLINDER
9727 034676 062701 000004    ADD    #4,R1              ;;BUMP R1 TO MR1
9728 034702 011011    MOV    (R0),(R1)          ;;LOAD MR1
9729 034704 013712 001600    MOV    L.CS1,(R2)        ;;LOAD CS1
9730 034710 012601    MOV    (SP)+,R1          ;;RESTORE REGISTER
9731 034712 012600    MOV    (SP)+,R0
9732 034714 000002    RTI                       ;;RETURN
9733
9734          .SBTTL  STORE RK611 REGISTERS
9735          ;*      ALL THE RK611 REGISTERS ARE STORED IN THE TEST TABLE T
9736          ;*      WITH THE EXCEPTION OF THE DATA BUFFER WHICH IS NOT STORED IN
9737          ;*      THIS ROUTINE.
9738
9739 034716 010046    GETRK: MOV    R0,-(SP)          ;;STORE REGISTERS TO BE USED
9740 034720 010146    MOV    R1,-(SP)
9741 034722 010346    MOV    R3,-(SP)
9742 034724 012700 001540    MOV    #T.CS1,R0          ;;SET POINTER TO TEST TABLE
9743 034730 010201    MOV    R2,R1              ;;SET POINTER TO RK611 BASE
9744 034732 012703 000012    MOV    #10,R3            ;;SET COUNT FOR 1ST 10 REGS
9745 034736 012120    1$:  MOV    (R1)+,(R0)+      ;;STORE RKCS1 THROUGH RKSPAR
9746 034740 005303    DEC    R3
9747 034742 001375    BNE   1$

```



```

9748 034744 062701 000002      ADD    #2,R1      ;BUMP POINTER PAST RKDB
9749 034750 005720      TST    (R0)+     ;BUMP POINTER PAST T.RKDB
9750 034752 012703 000004      MOV    #4,R3     ;SET COUNT FOR LAST 5 REGS
9751 034756 012120      2$:  MOV    (R1)+,(R0)+ ;STORE RKMRI THROUGH RKMRI
9752 034760 005303      DEC    R3
9753 034762 001375      BNE    2$
9754 034764 012603      MOV    (SP)+,R3  ;RESTORE REGISTERS
9755 034766 012601      MOV    (SP)+,R1
9756 034770 012600      MOV    (SP)+,R0
9757 034772 000002      RTI             ;RETURN
9758      ;SBTTL BIT COUNTER IN A WORD
9759      ;* THE WORD WHOSE BITS MUST BE COUNTED IS PLACED ON THE STACK
9760      ;* BY THE CALLING ROUTINE. THE NUMBER OF BITS FOUND IN THE WORD
9761      ;* ARE PASSED BACK ON THE STACK.
9762
9763 034774 016637 000002 001256 BITCNT: MOV    2(SP), $TMP16 ;GET WORD WHOSE BITS ARE TO BE COUNTED
9764 035002 010346      MOV    R3,-(SP) ;STORE R3
9765 035004 005037 001260      CLR    $TMP16   ;CLEAR $TMP16 FOR COUNTING
9766 035010 012703 000021      MOV    #17.,R3 ;SET A SHIFT COUNTER
9767 035014 000241      CLC             ;CLEAR CARRY
9768 035016 006037 001256      1$:  ROR    $TMP16   ;ROTATE WORD.
9769 035022 103407      BCS    3$      ;WAS BIT SHIFTED OUT A 1?
9770 035024 005303      2$:  DEC    R3      ;NO - DEC COUNT
9771 035026 001373      BNE    1$      ;LOOP IF NOT ZERO
9772 035030 012603      MOV    (SP)+,R3 ;RESTORE R3
9773 035032 013766 001260 000002      MOV    $TMP17,2(SP) ;PUT COUNT OF BITS ON STACK
9774 035040 000204      RTS    R4      ;RETURN
9775 035042 005237 001260      3$:  INC    $TMP17  ;BUMP COUNT
9776 035046 000766      BR     2$      ;LOOP
9777
9778      .SBTTL MAINTENANCE CLOCK ROUTINE
9779      ;* THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
9780      ;* ADDRESS AFTER THE CALL. THE FIRST BYTE CONTAINS THE NUMBER
9781      ;* OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
9782      ;* CLOCKED THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
9783      ;* CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
9784
9785      MCLK:
9786 035050 010046      MOV    R0,-(SP) ;:PUSH R0 ON STACK
9787 035052 010146      MOV    R1,-(SP) ;:PUSH R1 ON STACK
9788 035054 112400      MOVB  (R4)+,R0  ;:GET NUMBER OF CONTROLLER PHASE ADDRESSES
9789 035056 112401      MOVB  (R4)+,R1  ;:GET PARTIAL PHASE ADDRESS COUNT
9790
9791 035060 006300      ASL    R0      ;MULTIPLY PHASE ADDRESS COUNT BY 4
9792 035062 006300      ASL    R0
9793 035064 060100      ADD    R1,R0   ;ADD IN PARTIALS
9794 035066 052762 000400 000026 1$:  BIS    #MCLK,RKMR1(R2) ;SET CLOCK
9795 035074 042762 000400 000026      BIC    #MCLK,RKMR1(R2) ;CLEAR MCLK
9796 035102 005300      DEC    R0      ;DECREMENT COUNT
9797 035104 001370      BNE    1$      ;LOOP IF NOT ZERO
9798 035106 012601      MOV    (SP)+,R1 ;:POP STACK INTO R1
9799 035110 012600      MOV    (SP)+,R0 ;:POP STACK INTO R0
9800 035112 000204      RTS    R4
9801      .SBTTL READ AND SORT HEADERS
9802      ;* THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
9803      ;* THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN

```



```

9804      ;*      ASCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
9805      ;*      READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
9806      ;*      EACH READ OF THE DATA BUFFER.
9807      ;*
9808      ;*      CALL:   JSR      R4,RDSTHD
9809      ;*      TCHKOP  ;RETURN POINT IF CERR IN READ HDR
9810      ;*      ERROR   4 ;OR 5, 6, 7
9811      ;*      ERROR   13 ;RETURN IF DATA LATE IN DB UNLOAD
9812      ;*      ERROR   2  ;RETURN IF TO SLOW OR
9813      ;*      ;IF HDR 0 NOT FOUND
9814      ;*
9815      035114 104413      RDSTHD: SAVREG
9816      035116 032737 100000 001656 BIT #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
9817      035124 001402      BEQ 20$ ;NO - SKIP
9818      035126 005077 144544      CLR @KWLADD ;RESET INTERRUPT
9819      035132 012700 000026      20$: MOV #26,R0 ;PRESET FOR 26 SECTOR FORMAT
9820      035136 032737 010000 001600 BIT #CFMT,L.CS1 ;IS 24 SECTOR MODE SET?
9821      035144 001402      BEQ 1$ ;NO - SKIP
9822      035146 012700 000024      MOV #24,R0 ;ELSE CHANGE TO 24 SECTOR MODE
9823      035152 012701 060604      1$: MOV #IBUFF,R1 ;SET POINTER TO INPUT BUFFER
9824      035156 010005      MOV R0,R5 ;SAVE NUMBER OF SECTORS
9825      035160 010104      MOV R1,R4 ;SAVE IBUFF ADDRESS
9826      035162 010203      MOV R2,R3 ;SET UP POINTER TO RKDB
9827      035164 062703 000024      ADD #RKDB,R3
9828      035170 013762 001626 000010 MOV DRVNUM,RKCS2(R2) ;LOAD DRIVE NUM
9829      035176 013762 001614 000020 MOV L.DCYL,RKDCYL(R2) ;LOAD CYLINDER NUM
9830      035204 013762 001606 000006 MOV L.DA,RKDA(R2) ;LOAD TRACK AND SECTOR
9831
9832      035212 012737 000020 001654 2$: MOV #16,TIMCNT ;SET TIME COUNTER
9833      035220 005037 001662      CLR INTSET ;CLEAR INTERRUPT FLAG
9834      035224 005037 001660      CLR LCLKTK ;CLEAR TICK COUNTER
9835      035230 013762 001600 000000 MOV L.CS1,RKCS1(R2) ;LOAD COMMAND
9836
9837      035236 005737 001662      3$: TST INTSET ;TEST IF INT OCCURRED
9838      035242 001020      BNE 4$ ;YES - SKIP
9839      035244 004737 034214      JSR PC,MYTIME ;WAIT 1 MS
9840      035250 005737 001660      TST LCLKTK ;HAVE WE WAITED 16 MS?
9841      035254 001770      BEQ 3$ ;NO - LOOP ON WAIT
9842
9843      035256 062766 000006 000006 ADD #6,6(SP) ;SET RETURN FOR TO SLOW
9844      035264 104420      TGETRK ;GET RK REGS
9845      035266 012737 052427 001370 MOV #EM4,EM2N ;LOAD MESSAGE "TO SLOW/NOT COMPLETE"
9846      035274 012737 047042 001372 MOV #OPER24,DH2N ;LOAD COMMAND "READ HEADER" FOR REPORT
9847      035302 000466      BR 10$ ;SKIP
9848
9849      035304 005762 000000      4$: TST RKCS1(R2) ;TEST FOR CONTROLLER ERROR
9850      035310 100474      BMI 11$ ;YES - SKIP
9851
9852      035312 011324      MOV (R3),(R4)+ ;STORE HEADERS
9853      035314 011324      MOV (R3),(R4)+
9854      035316 011324      MOV (R3),(R4)+
9855
9856      035320 005762 000010      TST RKCS2(R2) ;TEST IF DATA LATE
9857      035324 100443      BMI 8$ ;YES - SKIP
9858
9859      035326 005300      DEC R0 ;DEC SECTOR COUNT

```



```

9860 035330 001330          BNE      2$          ;IF NOT ZERO - LOOP
9861
9862 035332 032737 100000 001656      BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
9863 035340 001403          BEQ      5$          ;NO - SKIP
9864 035342 012777 000100 144326      MOV      #BIT6,AKWLADD ;SET INTERRUPT ENABLE
9865 035350 032761 000037 000002 5$:   BIT      #37,2(R1)      ;HEADER AT TOP OF BUFF=HEAD 0?
9866 035356 001413          BEQ      6$          ;YES - SKIP
9867 035360 012124          MOV      (R1)+,(R4)+   ;ELSE MOV THIS HEADER TO BOTTOM
9868 035362 012124          MOV      (R1)+,(R4)+
9869 035364 012124          MOV      (R1)+,(R4)+
9870
9871 035366 005305          DEC      R5          ;TEST FO INSURE HEAD 0 IS FOUND
9872 035370 001367          BNE      5$          ;IF ALL HEADERS NOT CHECKED - LOOP
9873 035372 012737 055157 001370      MOV      #EM56,EM2N   ;ELSE "HEADER 0 NOT FOUND" MESSAGE
9874 035400 005037 001372          CLR      DH2N
9875 035404 000421          BR       9$          ;SKIP
9876
9877 035406 012700 060604          5$:   MOV      #IBUFF,RO    ;GET TOP OF IBUFF
9878 035412 012120          7$:   MOV      (R1)+,(RO)+  ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
9879 035414 012120          MOV      (R1)+,(RO)+
9880 035416 012120          MOV      (R1)+,(RO)+
9881 035420 020004          CMP      RO,R4
9882 035422 001373          BNE      7$          ;ALL HEADERS MOVED?
9883                                     ;NO - LOOP
9884 035424 062766 000010 000006      ADD      #10,6(SP)    ;SET UP FOR GOOD RETURN
9885 035432 000423          BR       11$
9886
9887 035434 012737 053442 001500 8$:   MOV      #EMDLT,EM13N ;"DATA LATE SET RESULT OF READ DB"
9888 035442 012737 055570 057702      MOV      #EMRDB,DF011A
9889 035450 062766 000004 000006 9$:   ADD      #4,6(SP)    ;SET ERROR RETURN
9890 035456 104420          TGETRK          ;GET RK REGS
9891 035460 013700 001302          10$:  MOV      $TESTN,RO   ;GET TEST NUMBER
9892 035464 006300          ASL      RO         ;SHIFT FOR INDEX
9893 035466 016037 032304 001264      MOV      $SW08TB(RO), $ESCAPE ;SET ESCAPE
9894 035474 162737 000002 001264      SUB      #2,$ESCAPE  ;TO NEXT SCOPE CALL
9895
9896 035502 104414          11$:  RESREG
9897 035504 000204          RTS      R4
9898
9899          .SBTTL  GET DRIVE STATUS
9900          ;*      THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
9901          ;*      THROUGH $REG17.  THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
9902          ;*      THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
9903          ;*      IN THE REGISTERS.
9904          ;*
9905          ;*CALL: JSR      R4,GETDRS
9906          ;*      BR       ERROR PROCESSING      ERROR RETURN
9907          ;*      BR       NO ERROR PROCESSING   GOOD RETURN
9908
9909 035506 104413          GETDRS: SAVREG
9910 035510 012762 100000 000000      MOV      #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
9911 035516 012700 001202          MOV      #$REG10,RO   ;PRESET ALL STATUS STORAGE TO
9912 035522 012701 000010          MOV      #8,R1        ;ALL ONES
9913 035526 012720 177777          1$:   MOV      #177777,(RO)+
9914 035532 005301          DEC      R1
9915 035534 001374          BNE      1$

```



```

9916 035536 012700 001206      MOV    #SREG12,RO      ;SET POINTER TO REG12 FOR A01 & B01
9917 035542 012701 000001      MOV    #1,R1          ;PRESET FOR PAIR ONE.
9918 035546 005037 001230      CLR    $TMP3          ;CLEAR ERROR SWITCH
9919 035552 013762 001610 000010 2$:  MOV    L,CS2,RKCS2(R2) ;LOAD DRIVE #
9920 035560 010162 000026      MOV    R1,RKMR1(R2)   ;LOAD MR1
9921 035564 012762 000001 000000      MOV    #BIT0,RKCS1(R2) ;DO SELECT
9922 035572 012703 000050      MOV    #40.,R3        ;WAIT FOR A FEW MICRO RECORDS TO
9923 035576 005303 3$:      DEC    R3             ;BIT SELECT FINISH
9924 035600 001376      BNE    3$
9925 035602 032762 100000 000000      BIT    #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
9926 035610 001415      BEQ    4$             ;NO - SKIP
9927 035612 032762 024000 000000      BIT    #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
9928 035620 001004      BNE    8$             ;YES - SKIP
9929 035622 032762 037400 000010      BIT    #37400,RKCS2(R2) ;TEST FOR ERRORS:
9930                                     ; NED!UPE!MDS!UFE!NEM!PGE
9931 035630 001405      BEQ    4$             ;NO - SKIP
9932 035632 012737 000001 001230 8$:  MOV    #1,$TMP3       ;SET ERROR FLAG
9933 035640 022020      CMP    (R0)+,(R0)+    ;BUMP TO LET THAT PAIR STAY ALL 1'S.
9934 035642 000404      BR     5$             ;SKIP
9935 035644 016220 000034 4$:  MOV    RKMR2(R2),(R0)+ ;STORE A WORD
9936 035650 016220 000036      MOV    RKMR3(R2),(R0)+ ;STORE B WORD
9937 035654 012762 100000 000000 5$:  MOV    #CLR,RKCS1(R2) ;CLEAR ANY OLD ERROR IN CONTROLLER
9938 035662 005701      TST    R1             ;IS R1 A 0 (LAST TRANSFER, PAIR 0)
9939 035664 001410      BEQ    6$             ;YES - SKIP
9940 035666 005201      INC    R1             ;ELSE BUMP TO NEXT PAIR
9941 035670 022701 000004      CMP    #4,R1          ;PAIR 3 JUST STORED?
9942 035674 001326      BNE    2$             ;NO - SKIP
9943 035676 005001      CLR    R1             ;ELSE SET TO PAIR 0
9944 035700 012700 001202      MOV    #SREG10,RO     ;PRESET POINTER FOR PAIR 0
9945 035704 000722      BR     2$             ;GO GET THEM
9946 035706 104414 6$:  RESREG ;EXIT HERE
9947 035710 005737 001230      TST    $TMP3          ;ANY ERROR IN STATUS GETTING
9948 035714 0010C1      BNE    7$             ;YES - SKIP
9949 035716 005724      TST    (R4)+          ;ELSE BUMP PART ERROR
9950 035720 000204 7$:  RTS    R4             ;RETURN
9951
9952 .SBTTL SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
9953 :;* THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
9954 :;* COMMAND. CERR AND DI ARE MONITORED FOR A SHORT
9955 :;* PEIROD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
9956 :;*
9957 :;* IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
9958 :;*
9959 :;* IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
9960 :;* ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
9961 :;* RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
9962 :;* SKIPS OVER THE ERROR CALL AND TEST ABORT.
9963 :;*
9964 :;* THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
9965 :;* AN ERROR MESSAGE AND BRANCH TO END OF TEST. THIS
9966 :;* IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
9967 :;* THE TEST.
9968
9969 035722 SSINIT:
9970 035722 010046      MOV    RO,-(SP)       ;;PUSH RO ON STACK
9971 035724 010146      MOV    R1,-(SP)       ;;PUSH R1 ON STACK

```



```

9972 035726 012701 000007      MOV      #7,R1          ;SET CLEAR COUNT
9973 035732 012700 001600      MOV      #L,CS1,RO     ;GET ADDRESS OF "L" REGS
9974 035736 012720 000100      MOV      #100,(RO)+    ;PRESET CS1
9975 035742 005020          7$: CLR      (RO)+      ;CLEAR THE NEXT
9976 035744 005301          DEC      R1           ;COUNT 0?
9977 035746 001375          BNE     7$           ;NO - LOOP
9978 035750 012762 000040 000010  MOV      #SCLR,RKCS2(R2);CLEAR SUBSYSTEM
9979 035756 012737 000012 001222  MOV      #10,$TMPD     ;SET A COUNTER
9980 035764 016237 000000 001540 1$: MOV      RKCS1(R2),T.CS1;GET CS1
9981 035772 032737 140000 001540  BIT      #CERR!DI,T.CS1;TEST IF ERROR OR DI SET
9982 036000 001433          BEQ     2$           ;NO - SKIP TO READY TEST
9983 036002 005337 001222          DEC      $TMPD        ;ELSE DECREMENT COUNTER
9984 036006 001366          BNE     1$           ;AND LOOP
9985 036010 032737 100000 001540  BIT      #CERR,T.CS1  ;TEST - IS IT CERR STILL SET
9986 036016 001404          BEQ     3$           ;NO - SKIP TO DI MESSAGE
9987 036020 012737 052522 001400  MOV      #EM5,EM3N     ;MESSAGE (SUBSYS CLR NOT RESET ERROR)
9988 036026 000403          BR      4$
9989 036030 012737 052566 001400 3$: MOV      #EM6,EM3N   ;MESSAGE (SUBSYS CLEAR NOT RESET DI)
9990 036036 104420          4$: TGETRK
9991 036040 013700 001302          MOV      $TESTN,RO    ;GET PRESENT TEST NUMBER
9992 036044 006300          ASL     RO           ;SHIFT FOR INDEX
9993 036046 016037 032304 001264  MOV      $$SWOBTBL(RO),$ESCAPE;LOAD ESCAPE TO ABORT TEST
9994 036054 162737 000002 001264  SUB     #2,$ESCAPE    ;SET TO NEXT SCOPE CALL
9995 036062 012601          MOV      (SP)+,R1     ;POP STACK INTO R1
9996 036064 012600          MOV      (SP)+,RO    ;POP STACK INTO RO
9997 036066 000414          BR      6$           ;SKIP TO EXIT
9998 036070 032737 000200 001540 2$: BIT      #RDY,T.CS1  ;TEST READY SET
9999 036076 001004          BNE     5$           ;YES - GOOD EXIT
10000 036100 012737 052645 001400  MOV      #EM7,EM3N    ;MESSAGE (SUBSYS CLR NOT SET READY)
10001 036106 000753          BR      4$
10002 036110 012601          5$: MOV      (SP)+,R1   ;RESTORE REGS
10003 036112 012600          MOV      (SP)+,RO
10004 036114 062716 000002          ADD     #2,(SP)      ;GOOD RETURN
10005 036120 000002          6$: RTI
10006
10007 .SBTTL WORD COUNT AT END OF OPERATION CHECK
10008 ;*
10009 ;* THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR
10010 ;* THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF UNEQUAL, THE
10011 ;* ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)
10012 ;*
10013 ;*CALL: JSR      R4,CHKWC
10014 ;* .WORD          ;EXPECTED WC VALUE
10015 036122 012437 046462          CHKWC: MOV      (R4)+,EXPWC ;STORE EXPECTED VALUE
10016
10017 036126 023737 046462 001542          CMP     EXPWC,T.WC    ;COMPARE
10018 036134 001406          BEQ     1$           ;EQUAL - SKIP
10019 036136 052737 000001 046520          BIS     #WCERR,GRP4ER;SET ERROR FLAG
10020 036144 013737 001542 046476          MOV      T.WC,REALWC ;STORE REAL WORD COUNT
10021 036152 000204          1$: RTS      R4      ;RETURN.
10022
10023 .SBTTL BUS ADDRESS AT END OF OPERATION CHECK
10024 ;*
10025 ;* THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF
10026 ;* A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE
10027 ;* INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.

```


D15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:09

MACY11 27(1006) 05-OCT-76 09:17 PAGE 185
BUS ADDRESS AT END OF OPERATION CHECK

SEQ 0185

```

10028      ;*      AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)
10029      ;*
10030      ;*      IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS
10031      ;*      ADDRESS IS THE STARTING BUS ADDRESS.
10032      ;*CALL: JSR      R4,CHKBA
10033
10034      036154 010046      CHKBA: MOV      RO,-(SP)
10035      036156 010146      MOV      R1,-(SP)
10036      036160 010346      MOV      R3,-(SP)
10037      036162 032737 000020 001610      BIT      #BAI,L.CS2      ;TEST IF BAI SET
10038      036170 001404      BEQ      4$              ;NO - SKIP
10039      036172 013737 001604 046466      MOV      L.BA,EXPBA      ;STORE EXPECTED BA (SAME AS STARTING BA)
10040      036200 000441      BR       3$
10041      036202 013700 001602      4$:  MOV      L.WC,RO      ;GET INITIAL WORD COUNT
10042      036206 005400      NEG      RO
10043      036210 113703 001601      MOVB     L.CS1+1,R3      ;GET BA16 & BA17
10044      036214 042703 177774      BIC      #177774,R3      ;CLEAR UNWANTED BITS
10045
10046      036220 005700      TST      RO      ;TEST IF INITIAL WORD COUNT 0
10047      036222 001003      BNE      6$          ;NO - SKIP
10048      036224 062703 000002      ADD      #2,R3      ;ADD 2 TO BA16,17 (65K WORD XFER)
10049      036230 000407      BR       9$          ;SKIP
10050      036232 005700      6$:  TST      RO      ;TEST IF INITIAL WC BIT 15 SET
10051      036234 100001      BPL      5$          ;NO - SKIP
10052      036236 005203      INC      R3          ;BUMP BA16,17 (32K WORD XFER)
10053      036240 006300      5$:  ASL      RO      ;SHIFT WORD COUNT TO MAKE MEM ADD CNT
10054      036242 063700 001604      ADD      L.BA,RO      ;ADD IN START BUFFER ADD
10055      036246 005503      ADC      R3          ;IF CARRY - ADD INTO BA16,17
10056      036250 013701 001542      9$:  MOV      T.WC,R1      ;GET END OF OPERATION WORD COUNT
10057      036254 001411      BEQ      1$          ;BRANCH IF ZERO
10058      036256 005401      NEG      R1
10059      036260 005701      TST      R1          ;TEST END OPERATION WC BIT 15 SET
10060      036262 100001      BPL      7$          ;NO - SKIP
10061      036264 005303      DEC      R3          ;DEC BA 16,17 (32K WC LEFT)
10062      036266 006301      7$:  ASL      R1          ;SHIFT WC TO MAKE MEM ADD CNT
10063      036270 160100      SUB      R1,RO      ;SUB FROM COMPUTED BUS ADDRESS
10064      036272 005603      SBC      R3          ;SUB CARRY FROM BA16,17
10065      036274 010337 046464      MOV      R3,EXPUBA      ;STORE EXPECTED UPPER BA BITS
10066      036300 010037 046466      1$:  MOV      RO,EXPBA
10067      036304 020037 001544      3$:  CMP      RO,T.BA      ;EQUAL TO COMPUTED?
10068      036310 001406      BEQ      2$          ;YES - SKIP
10069      036312 052737 000004 046520      BIS      #BAERR,GRP4ER  ;ELSE SET BAERR FLAG
10070      036320 013737 001544 046502      MOV      T.BA,REALBA    ;STORE REAL BUS ADDRESS
10071      036326 113703 001541      2$:  MOVB     T.CS1+1,R3      ;GET REAL UPPER BA
10072      036332 042703 177774      BIC      #177774,R3      ;CLEAR UNWANTED BITS
10073      036336 020337 046464      CMP      R3,EXPUBA      ;CHECK IF EQUAL
10074      036342 001405      BEQ      8$          ;YES - SKIP
10075      036344 052737 000002 046520      BIS      #UBAERR,GRP4ER ;ELSE SET UBA ERROR
10076      036352 010337 046500      MOV      R3,REALUB      ;STORE REAL UPPER BA
10077      036356 012603      8$:  MOV      (SP)+,R3
10078      036360 012601      MOV      (SP)+,R1
10079      036362 012600      MOV      (SP)+,RO
10080      036364 000204      RTS      R4
10081
10082      ;SBTTL  CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION
10083      ;*      THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA

```


E15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 186
CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION

SEQ 0186

```

10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096 036366 104413
10097 036370 013700 001602
10098 036374 005400
10099 036376 013701 001542
10100 036402 001401
10101 036404 005401
10102 036406 160100
10103 036410 005001
10104
10105
10106
10107
10108 036412 022700 000400
10109 036416 003004
10110 036420 005201
10111 036422 162700 000400
10112 036426 000771
10113 036430 005700
10114 036432 001401
10115 036434 005201
10116
10117
10118
10119 036436 012703 000026
10120 036442 032737 010000 001600
10121 036450 001402
10122 036452 012703 000024
10123
10124
10125 036456 013737 001614 046470
10126 036464 113704 001607
10127 036470 042704 177400
10128 036474 113705 001606
10129 036500 042705 177400
10130 036504 005301
10131 036506 005205
10132 036510 020503
10133 036512 001010
10134 036514 005005
10135 036516 005204
10136 036520 022704 000003
10137 036524 001003
10138 036526 005004
10139 036530 005237 046470

```

```

;* 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
CHKCTS: SAVREG
MOV L.WC,R0 ;GET SPECIFIED WORD COUNT
NEG R0 ;NEGATE IT
MOV T.WC,R1 ;GET END OF OPERATION WORD COUNT
BEQ 10$ ;IF ZERO - SKIP
NEG R1 ;NEGATE IT
10$: SUB R1,R0 ;COMPUTE ACTUAL WORDS TRANSFERRED
CLR R1 ;CLEAR R1 FOR COUNTING
;
; THE FOLLOWING CODE DETERMINES HOW MANY SECTORS OF DATA HAS BEEN
; TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE
; END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR
; IS CALCULATED.
1$: CMP #400,R0
BGT 2$
INC R1
SUB #400,R0
BR 1$
2$: TST R0
BEQ 3$
INC R1
;
; AT THIS POINT R1 HAS A COUNT OF THE
; NUMBER OF FULL SECTOR TRANSFER + 1 IF A
; PARTIAL SECTOR WAS TRANSFERRED.
3$: MOV #26,R3
BIT #CFMT,L.CS1
BEQ 4$
MOV #24,R3
;
; R3 HAS BEEN SET UP WITH THE NUMBER
; OF SECTORS IN A TRACK FOR THE FORMAT USED
4$: MOV L.DCYL,EXPCYL ;GET STARTING VALUES FOR CYLINDER
MOVB L.DT,R4 ;TRACK
BIC #177400,R4
MOVB L.DS,R5 ;SECTOR
BIC #177400,R5
5$: DEC R1 ;ADJUST COUNT FOR ZERO DETECT
INC R5 ;BUMP SECTOR COUNT
CMP R5,R3 ;DID THIS MAKE SECTOR COUNT > 1 TRACK?
BNE 6$ ;NO - SKIP
CLR R5 ;ELSE CLEAR SECTOR COUNT
INC R4 ;BUMP TRACK COUNT
CMP #3,R4 ;DID THIS MAKE TRK COUNT > 1 CYLINDER?
BNE 6$ ;NO - SKIP
CLR R4 ;ELSE CLEAR TRACK COUNT
INC EXPCYL ;BUMP CYLINDER COUNT

```


F15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:09

MACY11 27(1006) 05-OCT-76 09:17 PAGE 187
CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION

SEQ 0187

10140	036534	005301			6\$:	DEC	R1	;DEC COUNT
10141	036536	001363				BNE	5\$;IF ZERO - EXIT
10142	036540	010437	046474			MOV	R4, EXPTRK	;STORE EXPECTED TRACK
10143	036544	010537	046472			MOV	R5, EXPSEC	;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10144	036550	023737	001560	046470		CMP	T.DCYL, EXPCYL	;TEST IF CYLINDER OK
10145	036556	001403				BEQ	7\$;YES - SKIP
10146	036560	052737	000010	046520		BIS	#CYLERR, GRP4ER	;NO - SET ERROR FLAG
10147	036566	120437	001547		7\$:	CMPB	R4, T.DA+1	;TEST TRACK OK
10148	036572	001403				BEQ	8\$;YES - SKIP
10149	036574	052737	000020	046520		BIS	#TRKERR, GRP4ER	;NO - SET ERROR FLAG
10150	036602	120537	001546		8\$:	CMPB	R5, T.DA	;TEST SECTOR COUNT OK
10151	036606	001403				BEQ	9\$;YES - SKIP
10152	036610	052737	000040	046520		BIS	#SECERR, GRP4ER	;USE SET ERROR FLAG
10153	036616	012700	046476		9\$:	MOV	#REALWC, R0	
10154	036622	013720	001542			MOV	T.WC, (R0)+	;STORE REAL WORD COUNT
10155	036626	013720	001544			MOV	T.BA, (R0)+	;STORE REAL BUS ADDRESS
10156	036632	013720	001560			MOV	T.DCYL, (R0)+	;STORE REAL CYLINDER ADDRESS
10157	036636	113710	001547			MOVB	T.DA+1, (R0)	;STORE REAL TRACK ADDRESS
10158	036642	005720				TST	(R0)+	
10159	036644	113710	001546			MOVB	T.DA, (R0)	;STORE REAL SECTOR ADDRESS
10160	036650	104414				RESREG		
10161	036652	000204				RTS	R4	
10162								
10163						.SBTTL	OPERATION CHECK ROUTINE	
10164						.*	THIS IS WHERE ALL HARDWARE ERROR INDICATORS AND SOME SOFTWARE	
10165						.*	ERRORS ARE CHECKED. THE GENERAL PROCEDURE FLOW IS AS FOLLOWS:	
10166						.*	THE ROUTINE IS CALLED WITH A TRAP (TCHKOP). THE LOCATION	
10167						.*	FOLLOWING THE TRAP CALL WILL HAVE AN ERROR TRAP WHICH	
10168						.*	THE ROUTINE WILL BYPASS IF NO ERROR IS FOUND. IF AN	
10169						.*	ERROR IS DETECTED, THE ERROR TRAP CALL IS MODIFIED	
10170						.*	BY THIS ROUTINE SUCH THAT THE ERROR TABLE ITEM WILL	
10171						.*	BE THE PROPER ITEM FOR THE FORMAT REQUIRED BY THIS	
10172						.*	ERROR. THE ERROR TRAP WILL BE MADE EITHER ERROR 4, 5, 6,	
10173						.*	7, OR 10. REFER TO THE ERROR ITEM TABLE FOR A DESCRIPTION	
10174						.*	OF THE FORMAT AND WHICH ERRORS ARE DISPLAYED IN WHAT	
10175						.*	FORMAT.	
10176						.*		
10177	036654	104413				CHKWE:	SAVREG	
10178	036656	011600					MOV	(SP), R0 ;GET POINTER TO ERROR WORDS
10179	036660	012037	001242				MOV	(R0)+, \$TMP10 ;STORE EXPECTED ERROR GROUP 1
10180	036664	012037	001244				MOV	(R0)+, \$TMP11 ;STORE EXPECTED ERROR GROUP 2
10181	036670	012037	001246				MOV	(R0)+, \$TMP12 ;STORE EXPECTED ERROR GROUP 3
10182	036674	010016					MOV	R0, (SP) ;STORE RETURN
10183	036676	012737	177777	001250			MOV	#-1, \$TMP13 ;SET FLAG - EXPECTED ERROR
10184	036704	000403					BR	CHKST
10185								
10186	036706	104413				CHKOP:	SAVREG	
10187	036710	005037	001250				CLR	\$TMP13 ;RESET EXPECTED ERROR FLAG
10188								
10189	036714	104420				CHKST:	TGETRK	;GET 611 REGS IO TRAP
10190	036716	005037	046512				CLR	GRP1ER ;CLEAR ERROR FLAGS
10191	036722	005037	046514				CLR	GRP2ER
10192	036726	005037	046516				CLR	GRP3ER
10193	036732	005037	046520				CLR	GRP4ER
10194	036736	005037	046634				CLR	GPSUMF ;CLEAR SUMMARY FLAGS
10195	036742	032737	024000	001540			BIT	#CS1ERBIT, T.CS1 ;TEST IF ERROR SET IN CS1

10196	036750	001111			BNE	4\$:YES - SKIP
10197	036752	032737	177400	001550	BIT	#CS2ERBIT,T.CS2		:TEST IF ERROR SET IN CS2
10198	036760	001105			BNE	4\$:YES - SKIP
10199	036762	032737	000070	001552	BIT	#DSERBIT,T.DS		:TEST IF ERROR SET IN DS
10200	036770	001101			BNE	4\$:YES - SKIP
10201	036772	005737	001554		TST	T.ER		:TEST IF ERROR SET IN ER
10202	036776	001076			BNE	4\$:YES - SKIP
10203	037000	032737	100000	001540	BIT	#CERR,T.CS1		:COMBINED ERROR SET?
10204	037006	001405			BEQ	9\$:NO - SKIP
10205	037010	052737	100000	046512	BIS	#CERNER,GRP1ER		:SET ERROR FLAG IN GROUP 1
10206	037016	000137	037546		JMP	25\$:SKIP
10207								
10208								
10209								
10210								
10211								
10212								
10213								
10214								
10215								
10216								
10217	037022	005737	001250		9\$: TST	\$TMP13		:TEST IF ERROR EXPECTED
10218	037026	001402			BEQ	62\$:NO - SKIP
10219	037030	000137	037546		JMP	25\$:YES - JUMP
10220	037034	013700	001540		62\$: MOV	T.CS1,RO		:GET CS1
10221	037040	042700	177741		BIC	#177741,RO		:CHECK IF OPERATION IS READ DATA,
10222	037044	022700	000020		CMP	#20,RO		:WRITE DATA, OR WRITE CHECK. IF
10223	037050	002445			BLT	3\$:NOT, SKIP ALL CHECKING IN GROUP
10224	037052	022700	000030		CMP	#30,RO		:FOUR
10225	037056	003042			BGT	3\$		
10226	037060	004437	036122		JSR	R4,CHKWC		:CHECK WORD COUNT
10227	037064	000000			.WORD	0		:EXPECTED WORD COUNT
10228	037066	004437	036154		JSR	R4,CHKBA		:CHECK BUS ADDRESS
10229	037072	004437	036366		JSR	R4,CHKCTS		:CHECK CYL, TRACK, & SECTOR
10230	037076	005737	046520		TST	GRP4ER		:ANY GROUP 4 ERRORS?
10231	037102	001430			BEQ	3\$:NO - SKIP
10232	037104	016037	046636	057662	MOV	CMNDLB(RO),DF010A		:LOAD ADDRESS OF COMMAND MESSAGE
10233	037112	013700	046520		MOV	GRP4ER,RO		:PUT GROUP 4 ERROR FLAG IN RO
10234	037116	005001			CLR	R1		:CLEAR R1 FOR INDEX COUNTER
10235	037120	006200			1\$: ASR	RO		:SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
10236	037122	103402			BCS	2\$:BE REPORTED, REST ARE IGNORED
10237	037124	005720			TST	(RO)+		:WHEN AN ERROR BIT IS FOUND,
10238	037126	000774			BR	1\$:GET THE ERROR MESSAGE ASSOCIATED
10239	037130	016037	046522	001450	2\$: MOV	GRP4MS(RO),EM10N		:WITH IT AND SET ERROR TABLE ITEM TO
10240	037136	016037	046462	001202	MOV	EXPWC(RO),\$REG10		:POINT TO THE MESSAGE. LOAD REG10 & 11
10241	037144	016037	046476	001204	MOV	REALWC(RO),\$REG11		:WITH EXPECTED & IS VALUES
10242	037152	104414			RESREG			:RESTORE REGISTER
10243	037154	012776	000010	000000	MOV	#10,(SP)		:MAKE THE ERROR CALL POINT TO THE
10244	037162	000002			RTI			:RIGHT TABLE ENTRY & RETURN.
10245	037164				3\$: RESREG			
10246	037164	104414			ADD	#2,(SP)		:BUMP RETURN PAST ERROR
10247	037166	062716	000002		RTI			:RETURN
10248	037172	000002						
10249								
10250								
10251								

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.

H15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 189
OPERATION CHECK ROUTINE

SEQ 0189

10252	037174	012700	046512	4\$:	MOV	#GRP1ER, R0	;SET UP GENERAL REGISTER AS POINTER
10253	037200	012701	001540		MOV	#T.CS1, R1	;CS1
10254	037204	012703	001550		MOV	#T.CS2, R3	;CS2
10255	037210	012704	001552		MOV	#T.DS, R4	;DS
10256	037214	012705	001554		MOV	#T.ER, R5	;AND ER
10257							
10258	037220	051510			BIS	(R5), (R0)	;SET ALL BITS IN GRP1ER THAT
10259							;CORRESPOND TO ERROR BITS IN RKER
10260	037222	042710	120701		BIC	#ILF!ECH!BSE!HVRC!OPI!DCK, (R0)	; CLEAR ALL THAT DON'T BELONG GRP1
10261							
10262	037226	032711	020000		BIT	#SPAR, (R1)	;TEST IF SPAR SET
10263	037232	001402			BEQ	5\$;NO - SKIP
10264	037234	052710	000001		BIS	#SPARERR, (R0)	;SET SPAR ERROR FLAG
10265							
10266	037240	032714	000010	5\$:	BIT	#ACLO, (R4)	;TEST ACLO SET
10267	037244	001402			BEQ	6\$;NO - SKIP
10268	037246	052710	000100		BIS	#ACLOERR, (R0)	;SET ACLO ERROR FLAG
10269							
10270	037252	032714	000020	6\$:	BIT	#SPDLSS, (R4)	;TEST SPEED LOSS SET
10271	037256	001402			BEQ	7\$;NO - SKIP
10272	037260	052710	000200		BIS	#SPDERR, (R0)	;SET SPEED LOSS ERROR FLAG
10273							
10274	037264	032714	000040	7\$:	BIT	#DROT, (R4)	;TEST IF DROT SET
10275	037270	001402			BEQ	8\$;NO - SKIP
10276	037272	052710	000400		BIS	#DROTERR, (R0)	;SET DROT ERROR FLAG
10277							
10278	037276	032711	100000	8\$:	BIT	#CERR, (R1)	;TEST CERR ITSELF SET
10279	037302	001002			BNE	10\$;YES - SKIP
10280	037304	032710	020000		BIT	#NCERWE, (R0)	;SET NO CERR WITH ERROR FLAG
10281							
10282	037310	012700	046514	10\$:	MOV	#GRP2ER, R0	;SET POINTER TO GROUP 2 ERROR FLAGS
10283							
10284	037314	032715	000100		BIT	#ECH, (R5)	;TEST IF ECH SET
10285	037320	001402			BEQ	11\$;NO - SKIP
10286	037322	052710	000001		BIS	#ECHERR, (R0)	;SET ECH FLAG
10287							
10288	037326	032715	100000	11\$:	BIT	#DCK, (R5)	;TEST DCK SET
10289	037332	001402			BEQ	12\$;NO - SKIP
10290	037334	052710	000002		BIS	#DCKERR, (R0)	;SET DCK ERROR FLAG.
10291							
10292	037340	032713	040000	12\$:	BIT	#WCE, (R3)	;TEST WRITE CHECK ERROR
10293	037344	001402			BEQ	120\$;NO - SKIP
10294	037346	052710	000004		BIS	#WCKERR, (R0)	;SET WCE BIT
10295	037352	032713	100000	120\$:	BIT	#DLT, (R3)	;TEST DATA LATE
10296	037356	001402			BEQ	13\$;NO - SKIP
10297	037360	052710	000010		BIS	#DLTERR, (R0)	;SET DLT ERROR FLAG
10298							
10299	037364	032715	020000	13\$:	BIT	#OPI, (R5)	;TEST OPI SET
10300	037370	001402			BEQ	14\$;NO - SKIP
10301	037372	052710	000020		BIS	#OPIERR, (R0)	;SET OPI ERROR FLAG
10302							
10303	037376	032715	000400	14\$:	BIT	#HVRC, (R5)	;TEST HVRC SET
10304	037402	001402			BEQ	16\$;NO - SKIP
10305	037404	052710	000040		BIS	#HVRCERR, (R0)	;SET HVRC FLAG
10306							
10307	037410	032715	000200	16\$:	BIT	#BSE, (R5)	;TEST BSE ERROR FLAG

10308	037414	001402		BEQ	17\$;NO - SKIP
10309	037416	052710	000100	BIS	#BSERR, (R0)	;SET BSE FLAG
10310						
10311	037422	012700	046516	17\$:	MOV	#GRP3ER, R0 ;SET POINTER TO GROUP 3 FLAGS
10312						
10313	037426	032713	010000	BIT	#NED, (R3)	;TEST NED SET
10314	037432	001402		BEQ	18\$;NO - SKIP
10315	037434	052710	000001	BIS	#NEDERR, (R0)	;SET NED FLAG
10316						
10317	037440	032711	004000	18\$:	BIT	#CTO, (R1) ;TEST CTO SET
10318	037444	001402		BEQ	19\$;NO - SKIP
10319	037446	052710	000002	BIS	#CTOERR, (R0)	;SET CTO FLAG
10320						
10321	037452	032713	000400	19\$:	BIT	#UFE, (R3) ;TEST UFE SET
10322	037456	001402		BEQ	20\$;NO - SKIP
10323	037460	052710	000004	BIS	#UFERR, (R0)	;SET UFE FLAG
10324						
10325	037464	032713	001000	20\$:	BIT	#MDS, (R3) ;TEST MDS SET
10326	037470	001402		BEQ	21\$;NO - SKIP
10327	037472	052710	000010	BIS	#MDSERR, (R0)	;SET MDE FLAG
10328						
10329	037476	032713	002000	21\$:	BIT	#PGE, (R3) ;TEST PGE SET
10330	037502	001402		BEQ	22\$;NO - SKIP
10331	037504	052710	000020	BIS	#PGERR, (R0)	;SET PGE FLAG
10332						
10333	037510	032713	004000	22\$:	BIT	#NEM, (R3) ;TEST NEM SET
10334	037514	001402		BEQ	23\$;NO - SKIP
10335	037516	052710	000040	BIS	#NEMERR, (R0)	;SET NEM FLAG
10336						
10337	037522	032713	020000	23\$:	BIT	#UPE, (R3) ;TEST UPE SET
10338	037526	001402		BEQ	24\$;NO - SKIP
10339	037530	052710	000100	BIS	#UPERR, (R0)	;SET UPE FLAG
10340						
10341	037534	032715	000001	24\$:	BIT	#ILF, (R5) ;TEST ILF SET
10342	037540	001402		BEQ	25\$;NO - SKIP
10343	037542	052710	000200	BIS	#ILFERR, (R0)	;SET ILF FLAG.

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,
 TEMP11, & TEMP12 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.

10344				:		
10345				:		
10346				:		
10347				:		
10348				:		
10349				:		
10350				:		
10351				:		
10352				:		
10353				:		
10354				:		
10355				:		
10356	037546	005737	001250	25\$:	TST	\$TMP13 ;CHECK IF AN ERROR WAS EXPECTED
10357	037552	001423		BEQ	110\$;NO - SKIP
10358	037554	012704	046512	MOV	#GRP1ER, R4	;GET ADDRESS OF ERROR
10359	037560	012705	001242	MOV	#\$TMP10, R5	;GET ADDRESS OF EXPECTED ERRORS
10360						
10361	037564	011500		26\$:	MOV	(R5), R0 ;GET EXPECTED ERROR
10362	037566	011401		MOV	(R4), R1	;GET GROUP ERROR FLAGS
10363	037570	020001		CMP	R0, R1	;ARE THEY EQUAL?

10364	037572	001003		BNE	27\$;NO - SKIP
10365	037574	005000		CLR	R0		;CLEAR EXPECTED ED
10366	037576	005001		CLR	R1		;CLEAR OCCURED
10367	037600	000403		BR	28\$		
10368							
10369	037602	010003		27\$:	MOV	R0,R3	;STORE EXPECTED ERRORS
10370	037604	040100			BIC	R1,R0	;RESET EXPECTED THAT OCCURRED
10371	037606	040301			BIC	R3,R1	;RESET OCCURRED THAT EXPECTED
10372	037610	010025		28\$:	MOV	R0,(R5)+	;STORE EXPECTED THAT DID NOT OCCUR
10373	037612	010124			MOV	R1,(R4)+	;STORE OCCURRED THAT WERE NOT EXPECTED
10374	037614	022705	001250		CMP	#\$TMP13,R5	;ALL GROUPS CHECKED.
10375	037620	001361			BNE	26\$;NO - LOOP

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.

10403	037622	005004		110\$:	CLR	R4	;CLEAR COUNTERS
10404	037624	005005			CLR	R5	
10405	037626	012700	001224		MOV	#\$TMP1,R0	;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
10406	037632	012701	001226		MOV	#\$TMP2,R1	;WHERE ASCIZ ADDRESSES GO
10407	037636	012703	046634		MOV	#\$PSUMF,R3	;POINTERS TO GROUP SUMMARY FLAGS
10408	037642	012710	057642		MOV	#\$DF007A,(R0)	;PRESET FOR GRP3 ERR MESSAGE BUILD
10409	037646	012711	001442		MOV	#\$DH7N,(R1)	
10410	037652	013746	046516		MOV	GRP3ER,-(SP)	;GET GROUP 3 ERRORS, PUT ON STACK
10411	037656	004437	034774		JSR	R4,BITCNT	;GO COUNT NUMBER AT ERRORS
10412	037662	005716			TST	(SP)	;ANY ERRORS?
10413	037664	001403			BEQ	29\$;NO - SKIP
10414	037666	061605			ADD	(SP),R5	;ADD IN ERROR TOTAL
10415	037670	052713	000004		BIS	#\$GRP3ST,(R3)	;SET BIT TO INDICATE GROUP 3 ERROR
10416							
10417	037674	005726		29\$:	TST	(SP)+	;CLEAR OFF STACK
10418	037676	005737	001250		TST	#\$TMP13	;ERROR EXPECTED
10419	037702	001412			BEQ	31\$;NO - SKIP

10420	037704	013746	001246		MOV	\$TMP12,-(SP)	;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
10421	037710	004437	034774		JSR	R4,BITCNT	;COUNT NUMBER OF ERRORS.
10422	037714	005716			TST	(SP)	;WERE THERE ANY
10423	037716	001403			BEQ	30\$;NO - SKIP
10424	037720	052713	000040		BIS	#GP3NR,(R3)	;SET GROUP 3 NOT RECEIVED ERROR FLAG
10425	037724	061604			ADD	(SP),R4	;ADD COUNT TO TOTAL THESE
10426							
10427	037726	005726		30\$:	TST	(SP)+	;CLEAR OFF STACK
10428	037730	013746	046514	31\$:	MOV	GRP2ER,-(SP)	;GET GROUP 2 ERRORS FOR COUNTING
10429	037734	004437	034774		JSR	R4,BITCNT	;COUNT BITS
10430	037740	005716			TST	(SP)	;ANY SET?
10431	037742	001407			BEQ	32\$;NO - SKIP
10432	037744	052713	000002		BIS	#GRP2ST,(R3)	;SET FLAG FOR GROUP 2 ERRORS
10433	037750	061605			ADD	(SP),R5	;ADD INTO TOTAL
10434	037752	012710	057616		MOV	#DF006A,(R0)	;STORE ADDRESS FOR BUILDING REPORT
10435	037756	012711	001432		MOV	#DH6N,(R1)	
10436							
10437	037762	005726		32\$:	TST	(SP)+	;CLEAR OFF STACK
10438	037764	005737	001250		TST	\$TMP13	;ANY EXPECTED ERRORS
10439	037770	001416			BEQ	34\$;NO - SKIP
10440	037772	013746	001244		MOV	\$TMP11,-(SP)	;GET GROUP 2 NOT RECEIVED ERRORS
10441	037776	004437	034774		JSR	R4,BITCNT	;COUNT NUMBER OF BITS
10442	040002	005716			TST	(SP)	;ANY SET?
10443	040004	001407			BEQ	33\$;NO - SKIP
10444	040006	052713	000020		BIS	#GP2NR,(R3)	;SET FLAG FOR GROUP 2 NOT RECEIVED
10445	040012	061604			ADD	(SP),R4	;ADD INTO TOTAL
10446	040014	012710	057616		MOV	#DF006A,(R0)	;STORE ADDRESS FOR BUILDING REPORT
10447	040020	012711	001432		MOV	#DH6N,(R1)	
10448							
10449	040024	005726		33\$:	TST	(SP)+	;CLEAR OFF STACK
10450	040026	013746	046512	34\$:	MOV	GRP1ER,-(SP)	;GET GROUP 1 ERROR FLAGS
10451	040032	004437	034774		JSR	R4,BITCNT	;COUNT THE NUMBER OF BITS
10452	040036	005716			TST	(SP)	;ANY SET?
10453	040040	001407			BEQ	35\$;NO - SKIP
10454	040042	052713	000001		BIS	#GRP1ST,(R3)	;SET FLAG FOR GROUP 1 ERRORS SET
10455	040046	061605			ADD	(SP),R5	;ADD INTO TOTAL
10456	040050	012710	057532		MOV	#DF004A,(R0)	;LOAD ADDRESS FOR BUILDING REPORT
10457	040054	012711	001412		MOV	#DH4N,(R1)	
10458	040060	005726		35\$:	TST	(SP)+	;CLEAR OFF STACK
10459	040062	005737	001250		TST	\$TMP13	;ANY EXPECTED ERRORS?
10460	040066	001416			BEQ	60\$;NO - SKIP
10461	040070	013746	001242		MOV	\$TMP10,-(SP)	;GET GROUP 1 NO RECEIVED ERROR
10462	040074	004437	034774		JSR	R4,BITCNT	;COUNT # OF BITS
10463	040100	005716			TST	(SP)	;ANY SET?
10464	040102	001407			BEQ	36\$;NO - SKIP
10465	040104	052713	000010		BIS	#GP1NR,(R3)	;SET FLAG FOR GROUP 1 NOT RECEIVED
10466	040110	061604			ADD	(SP),R4	;ADD INTO TOTAL
10467	040112	012710	057532		MOV	#DF004A,(R0)	;LOAD ADDRESS FOR BUILDING REPORT
10468	040116	012711	001412		MOV	#DH4N,(R1)	
10469	040122	005726		36\$:	TST	(SP)+	;CLEAR OFF STACK.
10470	040124	032713	000011	60\$:	BIT	#GRP1ST!GP1NR,(R3)	;ANY GROUP 1 ERROR
10471	040130	001414			BEQ	52\$;NO - SKIP
10472	040132	042713	040000		BIC	#DRSTER,(R3)	
10473	040136	004437	035506		JSR	R4,GETDRS	
10474	040142	000401			BR	51\$;ERROR RETURN
10475	040144	000406			BR	52\$;NO ERROR RETURN

10476 040146 012710 057562
10477 040152 012711 001422
10478 040156 052713 040000
10479 040162
10480
10481
10482
10483
10484
10485
10486
10487
10488
10489
10490
10491
10492
10493
10494
10495
10496
10497
10498
10499
10500
10501
10502
10503
10504
10505
10506
10507 040162 032777 020000 140750
10508 040170 001402
10509 040172 000137 040650
10510 040176 005737 001250
10511 040202 001004
10512
10513
10514
10515 040204 012771 056536 000000
10516 040212 000411
10517 040214 012771 056357 000000
10518 040222 032713 000070
10519 040226 001003
10520 040230 012771 056454 000000
10521 040236 013701 001540
10522 040242 042701 177741
10523
10524 040246 016170 046636 000000
10525
10526 040254 032713 000007
10527 040260 001462
10528
10529
10530
10531 040262 013701 046516

51\$: MOV #DFO05A,(R0) ;CHANGE TO FORMAT 5 - STORE ADDRESS
MOV #DH5N,(R1) ;FOR BUILDING REPORT
BIS #DRSTER,(R3) ;SET DRIVE STATUS ERROR

52\$:

THE ERRORS ARE COUNTED, FLAGS SET TO INDICATE WHICH ERRORS ARE TO BE REPORTED, AND THE ERROR FORMAT HAS BEEN SELECTED. THE FOLLOWING CODE WILL TYPE ALL THE ERRORS, LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE ERROR ITEM TABLE AND LOAD THE PROPER HEADER MESSAGE ADDRESS IN THE PROPER DF 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

(R0)=#\$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,DSWR ;IS REPORT INHIBITED?

BEQ 37\$;NO - SKIP

JMP 53\$;ELSE EXIT

37\$: TST \$TMP13 ;WERE ERRORS EXPECTED?

BNE 38\$;YES - SKIP

; IF NO ERRORS WERE EXPECTED, \$TMP10,11, &12 ARE NOT MEANINGFUL

MOV #DH007,2(R1) ;HEADER = ERROR IN OPERATION

BR 39\$

38\$: MOV #DH005,2(R1) ;PRESET HDRMSG = EXPECTED NOT SET

BIT #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS?

BNE 39\$;YES - SKIP

MOV #DH006,2(R1) ;SET MESSAGE TO UNEXPECTED ERROR SET

39\$: MOV T.CS1,R1 ;GET CS1

BIC #177741,R1 ;CLEAR ALL BUT COMMAND

MOV CMNDLB(R1),2(R0) ;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


```

10532 040266 012700 046536          MOV      #GRP3MS,R0      ;SET POINTER TO GRP3 ERROR MESSAGES
10533 040272 005037 001252          CLR      $TMP14        ;CLEAR GROUP PRINTING INDICATOR
10534 040276 012737 000021 001254 40$:  MOV      #17.,$TMP15   ;PRESET SHIFT COUNT
10535 040304 000241                   CLC                      ;CLEAR CARRY
10536 040306 006001                   ROR      R1              ;ROTATE ERROR FLAGS
10537 040310 103406                   BCS     42$             ;WAS BIT SHIFTED OUT SET?
10538 040312 062700 000002          141$:  ADD      #2,R0      ;NO - BUMP POINTER
10539 040316 005337 001254          DEC      $TMP15        ;DEC SHIFT COUNT
10540 040322 001371                   BNE     41$            ;LOOP IF SHIFT NOT ZERO
10541 040324 000411                   BR      44$
10542 040326 011037 040340          42$:  MOV      (R0),43$   ;GET ERROR MESSAGE ADDRESS FROM TABLE
10543 040332 104401 001273          TYPE    , $CRLF        ;TYPE CRLF
10544 040336 104401                   TYPE    ;TYPE ERROR MESSAGE
10545 040340 000000          43$:  .WORD    ;MESSAGE ADDRESS GOES HERE
10546 040342 005305                   DEC     R5              ;DECEREMENT TOTAL ERROR COUNT.
10547 040344 001362                   BNE    141$            ;LOOP IF ZERO
10548 040346 000427                   BR     46$            ;ELSE EXIT GPR ERROR PRINT LOOP
10549
10550 040350 005713          44$:  TST     (R3)        ;TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
10551 040352 100455                   BMI    47$            ;YES - SKIP
10552 040354 005737 001252          TST     $TMP14        ;PRINTING GROUP 3?
10553 040360 001007                   BNE    45$            ;NO -SKIP
10554 040362 013701 046514          MOV     GRP2ER,R1     ;ELSE SET TO GROUP 2, GET GRP2ER
10555 040366 012700 046556          MOV     #GRP2MS,R0   ;& SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10556 040372 005237 001252          INC     $TMP14        ;BUMP TO INDICATE PRINTING GROUP 2
10557 040376 000737                   BR     40$            ;GO RESTART PRINT LOOP
10558 040400 022737 000002 001252 45$:  CMP     #2,$TMP14     ;PRINTING GROUP 1?
10559 040406 001407                   BEQ    46$            ;YES - EXIT GPR ERROR PRINT LOOP.
10560 040410 013701 046512          MOV     GRP1ER,R1     ;ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10561 040414 012700 046574          MOV     #GRP1MS,R0   ;SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10562 040420 005237 001252          INC     $TMP14        ;BUMP TO INDICATE PRINTING GROUP 1
10563 040424 000724                   BR     40$            ;RESTART PRINT LOOP.
10564
10565 040426 005737 001250          46$:  TST     $TMP13        ;EXPECTING ERRORS?
10566 040432 001452                   BEQ    49$            ;NO - SKIP
10567
10568 ;
10569 ;
10570 040434 005713          TST     (R3)        ;TEST IF PRINTING NOT RECEIVED ERRORS
10571 040436 100423                   BMI    47$            ;YES - SKIP
10572 040440 032713 000070          BIT     #GP1NR!GP2NR!GP3NR,(R3) ;ANY NOT RECEIVED ERRORS
10573 040444 001445                   BEQ    49$            ;NO - SKIP
10574 040446 032713 000007          BIT     #GRP1ST!GRP2ST!GRP3ST,(R3) ;ANY NOT RECEIVED ERRORS?
10575 040452 001404                   BEQ    146$          ;NO - SKIP LABEL FOR UNEXPECTED ERRORS
10576 040454 104401 001273          TYPE    , $CRLF        ;TYPE CRLF
10577 040460 104401 056454          TYPE    ,DH006        ;TYPE HEADER FOR PREVIOUS ERRORS
10578 040464 052737 100000 046634 146$:  BIS     #REPNR,GPSUMF ;SET PRINTING NOT RECEIVED ERRORS SWITCH
10579 040472 010405                   MOV     R4,R5         ;MOVE TOTAL ERRORS TO R5
10580 040474 013701 001246          MOV     $TMP12,R1     ;GET GRP3 NOT RECEIVED ERRORS
10581 040500 012700 046536          MOV     #GRP3MS,R0   ;SET POINTER TO GROUP 3 MESSAGES
10582 040504 000672                   BR     140$          ;GO START PRINT LOOP
10583 040506 005737 001252          47$:  TST     $TMP14        ;PRINTING GROUP 3?
10584 040512 001007                   BNE    48$            ;NO - SKIP
10585 040514 013701 001244          MOV     $TMP11,R1     ;ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10586 040520 012700 046556          MOV     #GRP2MS,R0   ;& SET POINTER TO GROUP 2 MESSAGE TABLE
10587 040524 005237 001252          INC     $TMP14        ;BUMP TO INDICATE GROUP 2 PRINTING

```



```

10588 040530 000662          BR      40$      ;GO START PRINT LOOP
10589 040532 022737 000002 001252 48$:  CMP      #2,$TMP14 ;PRINTING GROUP 1?
10590 040540 001407          BEQ      49$      ;YES - EXIT LOOP
10591 040542 017701 001242          MOV      $TMP10,R1 ;SET POINTER TO GROUP 1 MESSAGE
10592 040546 012700 046574          MOV      #GRP1MS,R0 ;TABLE AND GET GROUP 1 ERRORS.
10593 040552 005237 001252          INC      $TMP14    ;BUMP TO INDICATE GROUP 1 PRINTING
10594 040556 000647          BR      40$      ;START LOOP AGAIN.
10595
10596 040560 032713 000077          49$:  BIT      #77,(R3) ;TEST IF ANY ERRORS TO BE REPORTED
10597
10598
10599 040564 001004          BNE      61$      ;GRP1ST!GRP2ST!GRP3ST
10600 040566 104414          RESREG          ;GP1NR!GP2NR!GP3NR
10601 040570 062716 000002          ADD      #2,(SP)  ;YES - SKIP
10602 040574 000002          RTI           ;ELSE EXIT
10603
10604 040576 112776 000007 000000 61$:  MOVB     #7,$(SP) ;PRESET FOR GROUP 3 ERROR RETURN.
10605 040604 032713 000022          BIT      #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10606 040610 001403          BEQ      50$      ;NO - SKIP
10607 040612 112776 000006 000000          MOVB     #6,$(SP) ;ELSE SET FOR GROUP 2 ERROR RETURN
10608
10609 040620 032713 000011          50$:  BIT      #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10610 040624 001411          BEQ      53$      ;NO - SKIP
10611 040626 112776 000004 000000          MOVB     #4,$(SP) ;ELSE SET FOR GROUP 1 ERROR RETURN.
10612 040634 032713 040000          BIT      #DRSTER,(R3) ;CHECK IF ERROR GETTING DRIVE STATUS
10613 040640 001403          BEQ      53$      ;NO - SKIP
10614 040642 112776 000005 000000          MOVB     #5,$(SP) ;ELSE CHANGE RETURN FORM GROUP 1
10615
10616 040650 013700 001302          53$:  MOV      $TESTN,R0 ;SET UP $ESCAPE TO FORCE
10617 040654 006300          ASL      R0        ;ABORT TO PRESENT TEST AFTER
10618 040656 016037 032304 001264          MOV      $$W08TB(R0),$ESCAPE ;ERROR IS REPORTED
10619 040664 162737 000002 001264          SUB      #2,$ESCAPE ;BUT GO TO NEXT SCOPE STATEMENT
10620 040672 104414          RESREG
10621 040674 000002          RTI           ;RETURN
10622
10623
10624
10625
10626
10627
10628
10629
10630
10631
10632
10633
10634
10635
10636
10637
10638 040676 012637 001236          BDSRCK: MOV     (SP)+,$TMP6 ;STORE OLD R4 CONTENTS
10639 040702 010437 001240          MOV     R4,$TMP7   ;GET RETURN ADDRESS
10640 040706 011404          MOV     (R4),R4    ;GET POINTER TO FIELD TO BE CHECKED
10641 040710 005037 001234          CLR     $TMP5      ;CLEAR A COUNTER
10642 040714 005714          1$:  TST     (R4)       ;TEST IF FIELD HAS NO (OR NO MORE) ENTRIES
10643 040716 100417          BMI     4$         ;YES - EXIT

```

```

*****
;SBTTL 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, 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>
*****

```


10644	040720	023724	001614		CMP	L.DCYL,(R4)+	: IS THIS ENTRY FOR THIS CYLINDER?
10645	040724	001012			BNE	3\$: NO - SKIP
10646	040726	005204			INC	R4	: BUMP TO TRACK
10647	040730	123714	001607		CMPB	L.DT,(R4)	: IS ENTRY FOR THIS TRACK?
10648	040734	001005			BNE	2\$: NO - SKIP
10649	040736	005046			CLR	-(SP)	: CLEAR STACK LOCATION
10650	040740	114416			MOVB	-(R4),(SP)	: PUT SECTOR NUMBER ON STACK
10651	040742	005237	001234		INC	\$TMP5	: BUMP COUNTER
10652	040746	000401			BR	3\$: BRANCH
10653							
10654	040750	005304		2\$:	DEC	R4	: DECREMENT POINTER TO WORD ALIGN
10655	040752	005724		3\$:	TST	(R4)+	: BUMP TO NEXT ENTRY
10656	040754	000757			BR	1\$: TEST NEXT ENTRY
10657							
10658	040756	013746	001234	4\$:	MOV	\$TMP5,-(SP)	: PUT COUNT ON STACK
10659	040762	013746	001236		MOV	\$TMP6,-(SP)	: PUT OLD R4 CONTENTS BACK ON STACK
10660	040766	013704	001240		MOV	\$TMP7,R4	: SET UP RETURN
10661	040772	005724			TST	(R4)+	: BUMP PAST PARAMETER
10662	040774	000204			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.
    
```

10663
10664
10665
10666
10667
10668
10669
10670
10671
10672
10673
10674
10675
10676
10677
10678
10679
10680
10681
10682
10683
10684
10685
10686
10687
10688
10689
10690
10691
10692
10693
10694
10695
10696
10697
10698
10699

10700
10701
10702
10703
10704
10705
10706
10707
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
10736
10737
10738
10739
10740
10741
10742
10743
10744
10745
10746
10747
10748
10749
10750
10751
10752
10753
10754
10755

```

:*      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:

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 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:

```

MOV    R0,-(SP)      ;; PUSH R0 ON STACK
MOV    R1,-(SP)      ;; PUSH R1 ON STACK
MOV    R3,-(SP)      ;; PUSH R3 ON STACK
MOV    R5,-(SP)      ;; PUSH R5 ON STACK
MOV    (R4)+,R0      ; GET PARAMETER WORD

```

```

040776
040776 010046
041000 010146
041002 010346
041004 010546
041006 012400

```


E16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:09

MACY11 27(1006) 05-OCT-76 09:17 PAGE 199
DATA GENERATION AND COMPARE ROUTINE

SEQ 0199

10812	041244	011601			7\$:	MOV	(SP),R1	;GET 1ST BAD SECTOR NUMBER
10813	041246	006301				ASL	R1	;MULTIPLY SECTOR NUMBER BY 6 TO
10814	041250	006301				ASL	R1	;LOCATE SECTOR TO BE MARKED BAD
10815	041252	061601				ADD	(SP),R1	
10816	041254	062601				ADD	(SP)+,R1	
10817	041256	062701	000002			ADD	#2,R1	;ADD 2 FOR 2ND WORD THAT SECTOR
10818	041262	043761	001224	062604		BIC	\$TMP1,0BUFF(R1)	; CLEAR BIT FOR BAD SECTOR IN HDR
10819	041270	043761	001224	062606		BIC	\$TMP1,0BUFF+2(R1)	; CORRECT THE HVRC BIT
10820	041276	005305				DEC	R5	;DECREMENT BAD SECTOR COUNT
10821	041300	001361				BNE	7\$;LOOP IF NOT ZERO
10822								
10823	041302	032737	100000	001224	9\$:	BIT	#BIT15,\$TMP1	;WERE WE DOING BS FACTORY LIST?
10824	041310	001730				BEQ	10\$;NO - GO CHECK IF COMPARE MUST BE DONE
10825	041312	012737	040000	001224		MOV	#BIT14,\$TMP1	;ELSE SET BIT TO BE RESET IN BAD HDR
10826	041320	013737	001644	041236		MOV	BSS26P,6\$;PRESET POINTER FOR 26 SECTOR MODE
10827	041326	032737	010000	001600		BIT	#CFMT,L.CS1	;TEST IF WE ARE DOING 26 SECTOR MODE
10828	041334	001736				BEQ	8\$;YES - SKIP TO START CHECK
10829	041336	013737	001642	041236		MOV	BSS24P,6\$;CHANGE POINTER TOR 24 SECTOR MODE
10830	041344	000732				BR	8\$;SKIP TO START CHECK
10831								
10832								
10833	041346	012701	000102		11\$:	START OF COMPARE		
10834	041352	032737	010000	001600		MOV	#102,R1	;PRESET FOR 102 WORDS OF HEADER
10835	041360	001402				BIT	#CFMT,L.CS1	;CHECK IF 26 SECTOR MODE
10836	041362	012701	000074			BEQ	12\$;YES - SKIP
10837						MOV	#74,R1	;CHANGE TO 74 WORDS OF HEADER
10838	041366	012704	060604		12\$:	MOV	#IBUFF,R4	;SET START OF HEADERS TO BE COMPARED
10839	041372	012705	062604			MOV	#0BUFF,R5	;SET START OF GOOD HEADERS
10840	041376	005003				CLR	R3	;CLEAR COUNTER
10841	041400	032700	040000			BIT	#BIT14,R0	;IS THIS A CONTINUATION OF EARLIER COMPARE
10842	041404	001412				BEQ	13\$;NO - SKIP
10843	041406	013705	001666		28\$:	MOV	DESHLD,R5	;GET VALUES WHERE PREVIOUS CHECK STOPPED
10844	041412	013704	001670			MOV	SRCHLD,R4	; DESTINATION AND SOURCE
10845	041416	013703	001672			MOV	WRDNUM,R3	; WORD NUMBER IN ERROR
10846	041422	013701	001674			MOV	WRDCNT,R1	; WORD COUNT LEFT IN COMPARE
10847	041426	005701				TST	R1	;TEST IF WORD COUNT LEFT = 0
10848	041430	001417				BEQ	50\$;YES - EXIT
10849								
10850	041432	032700	030000		13\$:	BIT	#BIT12!BIT13,R0	;MEM MANAGE REQUIRED?
10851	041436	001402				BEQ	25\$;NO - SKIP
10852	041440	005237	177572			INC	@#SRO	;TURN IT ON
10853	041444	022425			25\$:	CMP	(R4)+,(R5)+	;COMPARE THE WORDS
10854	041446	001012				BNE	14\$;SKIP IF NOT EQUAL
10855	041450	005203				INC	R3	;BUMP WORD NUMBER IN ERROR
10856	041452	005301				DEC	R1	;DEC WORD COUNT LEFT IN COMPARE
10857	041454	001373				BNE	25\$;LOOP IF NOT ZERO
10858	041456	032700	030000			BIT	#BIT12!BIT13,R0	;MEM MANAGE IN USE?
10859	041462	001402				BEQ	50\$;NO - SKIP
10860	041464	005337	177572			DEC	@#SRO	;TURN IT OFF
10861	041470				50\$:			
10862	041470	012604				MOV	(SP)+,R4	;POP STACK INTO R4
10863	041472	000427				BR	16\$	
10864								
10865								
10866								
10867	041474	010537	001666		14\$:	MOV	R5,DESHLD	;STORE DESTINATION

F16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 200
DATA GENERATION AND COMPARE ROUTINE

SEG 0200

```

10868 041500 010437 001670      MOV      R4,SRCHLD      ; SOURCE
10869 041504 014537 001202      MOV      -(R5),$REG10  ; LOAD GOOD WORD FOR REPORT
10870 041510 014437 001204      MOV      -(R4),$REG11  ; BAD WORD
10871 041514 010337 001206      MOV      R3,$REG12     ; WORD NUMBER
10872 041520 005301              DEC      R1             ; DEC COUNT LEFT FOR CONTINUATION
10873 041522 005203              INC      R3             ; BUMP BAD WORD NUMBER
10874 041524 010137 001674      MOV      R1,WRDCNT     ; STORE COUNT LEFT
10875 041530 010337 001672      MOV      R3,WRDNUM     ; WORD NUM IN ERROR
10876 041534 032700 030000      BIT      #BIT12!BIT13,R0 ; MEM MANAGE IS USE?
10877 041540 001402              BEQ      15$           ; NO - SKIP
10878 041542 005337 177572      DEC      @#SR0         ; TURN IT OFF
10879
10880 041546              15$:
10881 041546 012604      MOV      (SP)+,R4      ; POP STACK INTO R4
10882 041550 005724      TST      (R4)+         ; ERROR RETURN
10883
10884 041552              16$:
10885 041552 012605      MOV      (SP)+,R5      ; POP STACK INTO R5
10886 041554 012603      MOV      (SP)+,R3      ; POP STACK INTO R3
10887 041556 012601      MOV      (SP)+,R1      ; POP STACK INTO R1
10888 041560 012600      MOV      (SP)+,R0      ; POP STACK INTO R0
10889 041562 000204      RTS      R4
10890
10891 ; DATA PATTERN PROCESSING ROUTINE
10892
10893 041564 032700 040000      17$: BIT      #BIT14,R0      ; CONTINUE WITH COMPARE?
10894 041570 001402      BEQ      29$           ; NO - SKIP
10895 041572 010446      MOV      R4,-(SP)      ; STORE RETURN
10896 041574 000704      BR       28$           ; GO CONTINUE COMPARE
10897
10898 041576 012705 062604      29$: MOV      #0BUFF,R5     ; GET DESTINATION
10899 041602 012703 060604      MOV      #IBUFF,R3     ; GET SOURCE
10900 041606 032700 030000      BIT      #BIT12!BIT13,R0 ; USE MEM MANAGE?
10901 041612 001412      BEQ      21$           ; NO - SKIP
10902
10903 041614 012437 172354      MOV      (R4)+,@#KIPAR6 ; LOAD PAR FOR RELOCATION
10904 041620 032700 010000      BIT      #BIT12,R0     ; RELOCATE SOURCE?
10905 041624 001403      BEQ      20$           ; NO - SKIP
10906 041626 012705 140070      MOV      #140070,R5    ; SET DESTINATION TO USE PAR6 + OFFSET
10907 041632 000402      BR       21$           ; SKIP
10908 041634 012703 140070      20$: MOV      #140070,R3    ; SET SOURCE TO USE PAR6 + OFFSET
10909
10910 041640 012401      21$: MOV      (R4)+,R1     ; STORE COUNT
10911 041642 010446      MOV      R4,-(SP)      ; STORE RETURN
10912 041644 010304      MOV      R3,R4         ; PUT IN Ibuff POINTER
10913 041646 005003      CLR      R3            ; CLEAR R3 FOR WORD NUMBER COUNTER
10914 041650 032700 000077      BIT      #77,R0        ; ANY DATA PATTERN SPECIFIED?
10915 041654 001666      BEQ      13$           ; NO - GO DO COMPARE
10916
10917 ; START OF GENERATION ROUTINE
10918
10919 041656 010537 001666      MOV      R5,DESHLD     ; STORE PARAMETERS FOR COMPARE
10920 041662 010437 001670      MOV      R4,SRCHLD
10921 041666 010337 001672      MOV      R3,WRDNUM
10922 041672 010137 001674      MOV      R1,WRDCNT
10923

```


G16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 201
DATA GENERATION AND COMPARE ROUTINE

SEQ 0201

```

10924      :      CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
10925      :      MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
10926      :      WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
10927
10928 041676 032700 030000      BIT      #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
10929 041702 001402      BEQ      33$      ;NO - SKIP
10930 041704 005237 177572      INC      @#SR0      ;TURN IT ON
10931 041710 032700 002000      33$: BIT      #BIT10,R0 ;GENERATE PATTERN IN Ibuff?
10932 041714 001403      BEQ      32$      ;NO - SKIP
10933 041716 010446      MOV      R4,-(SP) ;ELSE SWAP R4 AND R5
10934 041720 010504      MOV      R5,R4
10935 041722 012605      MOV      (SP)+,R5
10936
10937 041724 122700 000001      32$: CMPB     #1,R0      ;PATTERN 1 (ALL ZEROS)?
10938 041730 001004      BNE     55$      ;NO - SKIP
10939 041732 005025      30$: CLR      (R5)+    ;CLEAR WORD IN BUFF
10940 041734 005301      DEC     R1      ;DEC WORD COUNT
10941 041736 001375      BNE     30$      ;LOOP UNTIL WORD COUNT ZERO
10942 041740 000550      BR      22$      ;EXIT BUILD
10943
10944 041742 122700 000007      55$: CMPB     #7,R0      ;PATTERN 7 (ALL ONES)?
10945 041746 001005      BNE     56$      ;NO - SKIP
10946 041750 012725 177777      31$: MOV      #-1,(R5)+ ;LOAD WORD IN BUFF
10947 041754 005301      DEC     R1      ;DEC WORD COUNT
10948 041756 001374      BNE     31$      ;LOOP UNTIL WORD COUNT ZERO
10949 041760 000540      BR      22$      ;EXIT BUILD
10950
10951 041762 122700 000002      56$: CMPB     #2,R0      ;PATTERN 2 SET UP
10952 041766 001003      BNE     57$
10953 041770 012703 045662      MOV     #PAT02,R3
10954 041774 000504      BR      70$
10955
10956 041776 122700 000003      57$: CMPB     #3,R0      ;PATTERN 3 SET UP
10957 042002 001003      BNE     58$
10958 042004 012703 045722      MOV     #PAT03,R3
10959 042010 000476      BR      70$
10960
10961 042012 122700 000004      58$: CMPB     #4,R0      ;PATTERN 4 SET UP
10962 042016 001003      BNE     59$
10963 042020 012703 045762      MOV     #PAT04,R3
10964 042024 000470      BR      70$
10965
10966 042026 122700 000005      59$: CMPB     #5,R0      ;PATTERN 5 SET UP
10967 042032 001003      BNE     60$
10968 042034 012703 046022      MOV     #PAT05,R3
10969 042040 000462      BR      70$
10970
10971 042042 122700 000006      60$: CMPB     #6,R0      ;PATTERN 6 SET UP
10972 042046 001003      BNE     61$
10973 042050 012703 046062      MOV     #PAT06,R3
10974 042054 000454      BR      70$
10975
10976 042056 122700 000010      61$: CMPB     #10,R0     ;PATTERN 10 SET UP
10977 042062 001003      BNE     62$
10978 042064 012703 046122      MOV     #PAT10,R3
10979 042070 000446      BR      70$

```


H16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 202
DATA GENERATION AND COMPARE ROUTINE

SEQ 0202

10980							
10981	042072	122700	000011	62\$:	CMPB	#11,R0	;PATTERN 11 SET UP
10982	042076	001003			BNE	63\$	
10983	042100	012703	046162		MOV	#PAT11,R3	
10984	042104	000440			BR	70\$	
10985							
10986	042106	122700	000012	63\$:	CMPB	#12,R0	;PATTERN 12 SET UP
10987	042112	001003			BNE	64\$	
10988	042114	012703	046222		MOV	#PAT12,R3	
10989	042120	000432			BR	70\$	
10990							
10991	042122	122700	000013	64\$:	CMPB	#13,R0	;PATTERN 13 SET UP
10992	042126	001003			BNE	65\$	
10993	042130	012703	046262		MOV	#PAT13,R3	
10994	042134	000424			BR	70\$	
10995							
10996	042136	122700	000014	65\$:	CMPB	#14,R0	;PATTERN 14 SET UP
10997	042142	001003			BNE	66\$	
10998	042144	012703	046322		MOV	#PAT14,R3	
10999	042150	000416			BR	70\$	
11000							
11001	042152	122700	000015	66\$:	CMPB	#15,R0	;PATTERN 15 SET UP
11002	042156	001003			BNE	67\$	
11003	042160	012703	046362		MOV	#PAT15,R3	
11004	042164	000410			BR	70\$	
11005							
11006	042166	122700	000016	67\$:	CMPB	#16,R0	;PATTERN 16 SET UP
11007	042172	001003			BNE	68\$	
11008	042174	012703	046422		MOV	#PAT16,R3	
11009	042200	000402			BR	70\$	
11010							
11011	042202	012703	046422	68\$:	MOV	#PAT16,R3	;SET UP FOR 16
11012							
11013	042206	032700	004000	70\$:	BIT	#BIT11,R0	;FIRST WORD REPEAT?
11014	042212	001020			BNE	73\$;YES - SKIP
11015	042214	010446			MOV	R4,-(SP)	;STORE R4
11016	042216	010046			MOV	R0,-(SP)	;STORE R0
11017	042220	012700	000020		MOV	#16,R0	;PRESET COUNT FOR PATTERN LENGTH
11018	042224	010504			MOV	R5,R4	;STORE START OF BUFF
11019							
11020	042226	012325		71\$:	MOV	(R3)+,(R5)+	;MOV WORD TO BUFF
11021	042230	005301			DEC	R1	;DEC WORD COUNT
11022	042232	001405			BEQ	74\$;EXIT IF ZERO
11023	042234	005300			DEC	R0	;DEC PAT LENGTH COUNT
11024	042236	001373			BNE	71\$;LOOP IF NOT ZERO
11025							
11026	042240	012425		72\$:	MOV	(R4)+,(R5)+	;REPEAT PATTERN IN BUFFER
11027	042242	005301			DEC	R1	;DEC WORD COUNT
11028	042244	001375			BNE	72\$;LOOP UNTIL WORD COUNT ZERO
11029							
11030	042246	012600		74\$:	MOV	(SP)+,R0	;RESTORE R0
11031	042250	012604			MOV	(SP)+,R4	;RESTORE R4
11032	042252	000403			BR	22\$;EXIT BUILD
11033							
11034	042254	011325		73\$:	MOV	(R3),(R5)+	;MOV THE SAME WORD INTO BUFFER
11035	042256	005301			DEC	R1	;DEC WORD COUNT


```

11036 042260 001375          BNE      73$          ;LOOP UNTIL ZERO
11037
11038 042262 032700 030000 22$:  BIT      #BIT12!BIT13,R0 ;MEMORY MANAGEMENT REQUIRED?
11039 042266 001402          BEQ      34$          ;NO - SKIP
11040 042270 005337 177572    DEC      @#SR0        ;TURN OFF MEM MANAGEMENT
11041 042274 005700          TST      R0          ;IS COMPARE REQUIRED?
11042 042276 100012          BPL      23$          ;NO - SKIP
11043 042300 013705 001666    MOV      DESHLD,R5   ;RESTORE COMPARE PARAMETERS
11044 042304 013704 001670    MOV      SRCHLD,R4
11045 042310 013703 001672    MOV      WRDNUM,R3
11046 042314 013701 001674    MOV      WRDCNT,R1
11047 042320 000137 041432    JMP      13$          ;GO START COMPARE
11048 042324
11049 042324 012604          MOV      (SP)+,R4   ;:POP STACK INTO R4
11050 042326 000137 041552    JMP      16$          ;GO TO EXIT
11051
11052
11053 .SBTTL *****
11054 .SBTTL PHASE LOCK LOOP CLOCK ADJUSTMENT ROUTINE
11055 .;* THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
11056 .;* PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
11057 .;* PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
11058 .;* THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
11059 .;* RESETS DIAGNOSTIC MODE BIT IN MR1. INSTRUCTIONS ON WHERE TO
11060 .;* SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
11061 .;*
11062 .;* THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
11063 042332 104401 051565    ADJCLK: TYPE      ,OPRD19          ;TYPE ADJUSTMENT INSTRUCTIONS
11064
11065 042336 012762 000040 000010    MOV      #SCLR,RKCS2(R2) ;CLEAR SUBSYSTEM
11066 042344 013762 001626 000010    MOV      DRVNUM,RKCS2(R2) ;SELECT DRIVE
11067 042352 012762 000001 000000    MOV      #1,RKCS1(R2)
11068 042360 032762 000200 000000 5$:  BIT      #RDY,RKCS1(R2) ;WAIT FOR READY
11069 042366 001774          BEQ      5$
11070 042370 032737 100000 001656    BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
11071 042376 001402          BEQ      1$          ;NO - SKIP
11072 042400 005077 137272    CLR      @KWLADD     ;CLEAR INTERRUPT ENABLE
11073
11074 042404 012762 000040 000026 1$:  MOV      #DMD,RKMR1(R2) ;SET DIAG MODE
11075 042412 012701 000014          MOV      #12.,R1     ;SET A COUNT
11076 042416 005301          DEC      R1          ;DEC COUNT
11077 042420 001376          BNE      2$          ;LOOP UNTIL ZERO
11078 042422 012762 000000 000026    MOV      #0,RKMR1(R2) ;CLEAR MR1
11079 042430 012701 000014          MOV      #12.,R1     ;SET COUNT
11080 042434 005301          DEC      R1          ;DEC COUNT
11081 042436 001376          BNE      3$          ;LOOP UNTIL ZERO
11082 042440 000761          BR       1$          ;RESTART LOOP
11083
11084 .SBTTL CONTROLLED HALT SUBROUTINE
11085 .;* THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
11086 .;* SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
11087 .;* NECESSARY, CERTAIN CYLINDERS ARE REFORMATTED. THE REFORMATTING
11088 .;* IS CONTROLLED BY THE LOCATION REFMT WHICH CONTAINS THE ADDRESS
11089 .;* OF THE CYLINDER TO BE REFORMATTED.
11090
11091 042442 012737 000112 001302 CTRHLT: MOV      #STN,$TESTN          ;SET UP FOR HALT FAIL

```


11092							
11093	042450	104416		TSSINIT			;CLEAR SUBSYSTEM
11094	042452	104003		ERROR 3			;BAD INIT ERROR
11095							
11096	042454	113700	001102	MOVB	\$TSTNM,RO		;GET CURRENT TEST NUMBER
11097	042460	042700	177400	BIC	#177400,RO		;CLEAR UNUSED BITS
11098	042464	022700	000003	CMP	#3,RO		;TEST IF TEST NUMBER 3
11099	042470	001464		BEQ	PROGEND		;GO TO HALT PROG
11100	042472	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
11101	042476	000113		RECAL			;RECAL
11102	042500	000000		0			;0 WORDS
11103	042502	000000		0			;0 IS BUFF ADDRESS
11104	042504	000		.BYTE	0		;SECTOR 0
11105	042505	000		.BYTE	0		;TRACK 0
11106	042506	000000		0			;CYLINDER 0
11107							
11108	042510	104417		TLOADRK			;LOAD RK REGS
11109	042512	104423		TWAT16			;WAIT FOR INTERRUPT
11110	042514	104002		ERROR 2			;TO SLOW/NOT COMPLETE ERROR
11111							
11112	042516	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11113	042520	104004		ERROR 4 ;OR 5, 6, 7			;REPORT ALL ERRORS
11114							
11115	042522	005037	001662	CLR	INTSET		;CLEAR INTERRUPT FLAG
11116	042526	104437		TWAT8S			;WAIT FOR SECOND INTERRUPT
11117	042530	104002		ERROR 2			
11118							
11119	042532	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11120	042534	104004		ERROR 4 ;OR 5, 6, 7			;REPORT ALL ERRORS
11121							
11122	042536	104416		TSSINIT			;CLEAR SUBSYSTEM
11123	042540	104003		ERROR 3			;BAD INIT ERROR
11124							
11125	042542	005737	001664	TST	REFMT		;TEST IF REFORMAT REQUIRED
11126	042546	001435		BEQ	PROGEND		;NO - GO TO HALT
11127	042550	104401	051730	TYPE	,OPR020		;TYPE MESSAGE
11128							
11129	042554	004437	034574	JSR	R4,LRLOAD		;LOAD "L" REGS
11130	042560	000127		WRHEAD			;WRHEAD
11131	042562	177676		-102			;-102 WORDS
11132	042564	062604		OBUFF			;OBUFF IS BUFF ADDRESS
11133	042566	000		.BYTE	0		;SECTOR 0
11134	042567	000		.BYTE	0		;TRACK 0
11135	042570	000312		312			;CYLINDER 312
11136							
11137	042572	005737	001664	TST	REFMT		;TEST IF CYL 0
11138	042576	100002		BPL	5\$;NO - SKIP
11139	042600	005037	001614	CLR	L.DCYL		;ELSE LOAD FOR CYL 0
11140	042604	004437	040776	JSR	R4,GENCOM		;GENERATE HEADERS
11141	042610	001200		1200			
11142							
11143	042612	104417		TLOADRK			;LOAD RK REGS
11144	042614	104434		TWAT159			;WAIT FOR INTERRUPT
11145	042616	104002		ERROR 2			;TO SLOW/NOT COMPLETE ERROR
11146							
11147	042620	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS

5\$:


```

11148 042622 104004          ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
11149
11150 042624 122737 000002 001607      CMPB  #2,L.DT      ;TEST IF TRACK 2 FORMATTED
11151 042632 001403          BEQ   PROGEN      ;YES - SKIP
11152 042634 105237 001607      INCB  L.DT        ;ELSE BUMP TRACK
11153 042640 000761          BR    5$          ;DO NEXT TRACK
11154
11155 042642 104401 052024      PROGEN:  TYPE      OPRO21 ;TYPE HALT MESSAGE
11156 042646 012706 001100      MOV   #STACK,SP  ;CLEAR STACK
11157 042652 105037 001103      CLRB $EFLG      ;CLEAR ERROR FLAG
11158 042656 005037 001264      CLR  $ESCAPE    ;CLEAR ESCAPE
11159 042662 005737 000042      TST  @#42       ;TEST IF MONITOR PRESENT
11160 042666 001404          BEQ   10$        ;NO - SKIP
11161 042670 005037 031270      CLR  $EOPCT    ;SET FOR END OF PROGRAM
11162 042674 000137 031242      JMP  $EOP       ;GO TO END OF PASS
11163
11164 042700 000000          10$:  HALT        ;HALT PROGRAM
11165 042702 000137 001744      JMP  START1     ;GO TO RESTART IF CONTINUE
11166
11167      .SBTTL HALT FAIL ROUTINE
11168      ;* THIS ROUTINE IS ENTERED IF A HARDWARE ERROR IS DETECTED WHEN
11169      ;* THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
11170 042706 000240          ABTFAIL:  NOP
11171 042710 104401 052063      TYPE  OPRO22    ;TYPE HALT FAIL MESSAGE
11172 042714 000137 042642      JMP  PROGEN     ;GO STOP PROGRAM
11173
11174      .SBTTL TYPE ROUTINE
11175
11176      ;*****
11177      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11178      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11179      ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11180      ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11181      ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11182      ;*
11183      ;*CALL:
11184      ;*1) USING A TRAP INSTRUCTION
11185      ;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11186      ;*OR
11187      ;* TYPE
11188      ;* MESADR
11189      ;*
11190 042720 105737 001157      $TYPE:  TSTB  $TPFLG ;; IS THERE A TERMINAL?
11191 042724 100002          BPL  1$         ;; BR IF YES
11192 042726 000000          HALT ;; HALT HERE IF NO TERMINAL
11193 042730 000430          BR   3$         ;; LEAVE
11194 042732 010046          1$:  MOV   RO, -(SP) ;; SAVE RO
11195 042734 017600 000002      MOV   @2(SP),RO ;; GET ADDRESS OF ASCIZ STRING
11196 042740 122737 000001 001316      CMPB  #APTENV,$ENV ;; RUNNING IN APT MODE
11197 042746 001011          BNE  62$        ;; NO GO CHECK FOR APT CONSOLE
11198 042750 132737 000100 001317      BITB  #APTSPool,$ENVM ;; SPOOL MESSAGE TO APT
11199 042756 001405          BEQ  62$        ;; NO GO CHECK FOR CONSOLE
11200 042760 010037 042770      MOV   RO,61$    ;; SETUP MESSAGE ADDRESS FOR APT
11201 042764 004737 032540      JSR  PC,$ATY3   ;; SPOOL MESSAGE TO APT
11202 042770 000000          61$: .WORD 0 ;; MESSAGE ADDRESS
11203 042772 132737 000040 001317      62$:  BITB  #APTCSUP,$ENVM ;; APT CONSOLE SUPPRESSED

```


M16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 207
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0207

```

11260          ;*CALL:
11261          ;*   MOV     NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
11262          ;*   TYPDS          ;;GO TO THE ROUTINE
11263
11264 043202          $TYPDS:
11265 043202 010046      MOV     R0,-(SP)      ;;PUSH R0 ON STACK
11266 043204 010146      MOV     R1,-(SP)      ;;PUSH R1 ON STACK
11267 043206 010246      MOV     R2,-(SP)      ;;PUSH R2 ON STACK
11268 043210 010346      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
11269 043212 010546      MOV     R5,-(SP)      ;;PUSH R5 ON STACK
11270 043214 012746 020200  MOV     #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
11271 043220 016605 000020  MOV     20(SP),R5    ;;GET THE INPUT NUMBER
11272 043224 100004          BPL     1$          ;;BR IF INPUT IS POS.
11273 043226 005405          NEG     R5          ;;MAKE THE BINARY NUMBER POS.
11274 043230 112766 000055 000001  MOVVB  #'-,1(SP)    ;;MAKE THE ASCII NUMBER NEG.
11275 043236 005000          CLR     R0          ;;ZERO THE CONSTANTS INDEX
11276 043240 012703 043416      MOV     #SDBLK,R3   ;;SETUP THE OUTPUT POINTER
11277 043244 112723 000040      MOVVB  #'',(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
11278 043250 005002          CLR     R2          ;;CLEAR THE BCD NUMBER
11279 043252 016001 043406      MOV     $DTBL(R0),R1 ;;GET THE CONSTANT
11280 043256 160105          SUB     R1,R5       ;;FORM THIS BCD DIGIT
11281 043260 002402          BLT    4$          ;;BR IF DONE
11282 043262 005202          INC     R2          ;;INCREASE THE BCD DIGIT BY 1
11283 043264 000774          BR     3$
11284 043266 060105          4$:   ADD     R1,R5    ;;ADD BACK THE CONSTANT
11285 043270 005702          TST    R2          ;;CHECK IF BCD DIGIT=0
11286 043272 001002          BNE    5$          ;;FALL THROUGH IF 0
11287 043274 105716          TSTB   (SP)        ;;STILL DOING LEADING 0'S?
11288 043276 100407          BMI    7$          ;;BR IF YES
11289 043300 106316          5$:   ASLB   (SP)        ;;MSD?
11290 043302 103003          BCC    6$          ;;BR IF NO
11291 043304 116663 000001 177777  MOVVB  1(SP),-1(R3)  ;;YES--SET THE SIGN
11292 043312 052702 000060          BIS    #'0,R2      ;;MAKE THE BCD DIGIT ASCII
11293 043316 052702 000040          7$:   BIS    #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
11294 043322 110223          MOVVB  R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
11295 043324 005720          TST    (R0)+      ;;JUST INCREMENTING
11296 043326 020027 000010          CMP    R0,#10     ;;CHECK THE TABLE INDEX
11297 043332 002746          BLT    2$          ;;GO DO THE NEXT DIGIT
11298 043334 003002          BGT    8$          ;;GO TO EXIT
11299 043336 010502          MOV    R5,R2      ;;GET THE LSD
11300 043340 000764          BR     6$          ;;GO CHANGE TO ASCII
11301 043342 105726          8$:   TSTB   (SP)+     ;;WAS THE LSD THE FIRST NON-ZERO?
11302 043344 100003          BPL    9$          ;;BR IF NO
11303 043346 116663 177777 177776  MOVVB  -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
11304 043354 105013          9$:   CLRB   (R3)      ;;SET THE TERMINATOR
11305 043356 012605          MOV    (SP)+,R5   ;;POP STACK INTO R5
11306 043360 012603          MOV    (SP)+,R3   ;;POP STACK INTO R3
11307 043362 012602          MOV    (SP)+,R2   ;;POP STACK INTO R2
11308 043364 012601          MOV    (SP)+,R1   ;;POP STACK INTO R1
11309 043366 012600          MOV    (SP)+,R0   ;;POP STACK INTO R0
11310 043370 104401 043416      TYPE   $SDBLK      ;;NOW TYPE THE NUMBER
11311 043374 016666 000002 000004  MOV    2(SP),4(SP) ;;ADJUST THE STACK
11312 043402 012616          MOV    (SP)+,(SP)
11313 043404 000002          RTI
11314 043406 023420          $DTBL: 10000.
11315 043410 001750          1000.

```


11316	043412	000144	
11317	043414	000012	
11318	043416	000004	
11319			
11320			
11321			
11322			
11323			
11324			
11325			
11326			
11327			
11328			
11329			
11330			
11331			
11332			
11333			
11334			
11335			
11336			
11337			
11338			
11339			
11340			
11341			
11342			
11343			
11344	043426	017646	000000
11345	043432	116637	000001 043651
11346	043440	112637	043653
11347	043444	062716	000002
11348	043450	000406	
11349	043452	112737	000001 043651
11350	043460	112737	000006 043653
11351	043466	112737	000005 043650
11352	043474	010346	
11353	043476	010446	
11354	043500	010546	
11355	043502	113704	043653
11356	043506	005404	
11357	043510	062704	000006
11358	043514	110437	043652
11359	043520	113704	043651
11360	043524	016605	000012
11361	043530	005003	
11362	043532	006105	
11363	043534	000404	
11364	043536	006105	
11365	043540	006105	
11366	043542	006105	
11367	043544	010503	
11368	043546	006103	
11369	043550	105337	043652
11370	043554	100016	
11371	043556	042703	177770

```

100.
10.
$DBLK: .BLKW 4
.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOS   N              ;;CALL FOR TYPEOUT
*   .BYTE  M              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*   .BYTE  M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPON   N              ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOC   N              ;;CALL FOR TYPEOUT

$TYPOS: MOV     2(SP),-(SP)    ;;PICKUP THE MODE
        MOVVB  1(SP),$OFILL  ;;LOAD ZERO FILL SWITCH
        MOVVB  (SP)+,$SOMODE+1 ;;NUMBER OF DIGITS TO TYPE
        ADD    #2,(SP)      ;;ADJUST RETURN ADDRESS
        BR     $TYPON
$TYPOC: MOVVB  #1,$OFILL    ;;SET THE ZERO FILL SWITCH
        MOVVB  #6,$SOMODE+1 ;;SET FOR SIX(6) DIGITS
$TYPON: MOVVB  #5,$SOCNT    ;;SET THE ITERATION COUNT
        MOV    R3,-(SP)     ;;SAVE R3
        MOV    R4,-(SP)     ;;SAVE R4
        MOV    R5,-(SP)     ;;SAVE R5
        MOVVB  $SOMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
        NEG    R4
        ADD    #6,R4        ;;SUBTRACT IT FOR MAX. ALLOWED
        MOVVB  R4,$SOMODE  ;;SAVE IT FOR USE
        MOVVB  $OFILL,R4   ;;GET THE ZERO FILL SWITCH
        MOV    12(SP),R5   ;;PICKUP THE INPUT NUMBER
        CLR    R3          ;;CLEAR THE OUTPUT WORD
1$:     ROL    R5           ;;ROTATE MSB INTO "C"
        BR    3$           ;;GO DO MSB
2$:     ROL    R5           ;;FORM THIS DIGIT
        ROL    R5
        ROL    R5
        MOV    R5,R3
3$:     ROL    R3           ;;GET LSB OF THIS DIGIT
        DECB  $SOMODE     ;;TYPE THIS DIGIT?
        BPL   7$          ;;BR IF NO
        BIC  #177770,R3  ;;GET RID OF JUNK

```



```

11372 043562 001002      BNE      4$      :: TEST FOR 0
11373 043564 005704      TST      R4      :: SUPPRESS THIS 0?
11374 043566 001403      BEQ      5$      :: BR IF YES
11375 043570 005204      4$: INC      R4      :: DON'T SUPPRESS ANYMORE 0'S
11376 043572 052703 000060      BIS      #'0,R3  :: MAKE THIS DIGIT ASCII
11377 043576 052703 000040      5$: BIS      #' R3  :: MAKE ASCII IF NOT ALREADY
11378 043602 110337 043646      MOV      R3,8$   :: SAVE FOR TYPING
11379 043606 104401 043646      TYPE     8$      :: GO TYPE THIS DIGIT
11380 043612 105337 043650      7$: DECB     $OCNT  :: COUNT BY 1
11381 043616 003347      BGT      2$      :: BR IF MORE TO DO
11382 043620 002402      BLT      6$      :: BR IF DONE
11383 043622 005204      INC      R4      :: INSURE LAST DIGIT ISN'T A BLANK
11384 043624 000744      BR       2$      :: GO DO THE LAST DIGIT
11385 043626 012605      6$: MOV      (SP)+,R5  :: RESTORE R5
11386 043630 012604      MOV      (SP)+,R4  :: RESTORE R4
11387 043632 012603      MOV      (SP)+,R3  :: RESTORE R3
11388 043634 016666 000002 000004      MOV      2(SP),4(SP)  :: SET THE STACK FOR RETURNING
11389 043642 012616      MOV      (SP)+,(SP)
11390 043644 000002      RTI      :: RETURN
11391 043646      8$: .BYTE     0      :: STORAGE FOR ASCII DIGIT
11392 043647      .BYTE     0      :: TERMINATOR FOR TYPE ROUTINE
11393 043650      .BYTE     0      :: OCTAL DIGIT COUNTER
11394 043651      .BYTE     0      :: ZERO FILL SWITCH
11395 043652 000000      $OMODE: .WORD    0  :: NUMBER OF DIGITS TO TYPE
11396
11397
11398
11399
11400 043654 000000      .ENABL   LSB
11401 043656 000000      $TKCNT: .WORD    0  :: NUMBER OF ITEMS IN QUEUE
11402 043660 000000      $TKQIN: .WORD    0  :: INPUT POINTER
11403 043662 000001      $TKQOUT: .WORD   0  :: OUTPUT POINTER
11404      043663      $TKQSRT: .BLKB   1  :: TTY KEYBOARD QUEUE
11405      043664      $TKQEND=.
11406
11407
11408      *TK INITIALIZE ROUTINE
11409      *THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
11410      *SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
11411
11412      *CALL:
11413      *      JSR      PC,$TKINT
11414      *      RETURN
11415
11415 043664 005037 043654      $TKINT: CLR      $TKCNT  :: CLEAR COUNT OF ITEMS IN QUEUE
11416 043670 012737 043662 043656      MOV      #$TKQSRT,$TKQIN  :: MOVE THE STARTING ADDRESS OF THE
11417 043676 013737 043656 043660      MOV      $TKQIN,$TKQOUT  :: QUEUE INTO THE INPUT & OUTPUT POINTERS.
11418 043704 012737 043734 000060      MOV      #$TKSRV,$TKVEC  :: INITIALIZE THE KEYBOARD VECTOR
11419 043712 012737 000200 000062      MOV      #200,$TKVEC+2  :: "BR" LEVEL 4
11420 043720 005777 135222      TST      $TKB          :: CLEAR DONE FLAG
11421 043724 012777 000100 135212      MOV      #100,$TKS      :: ENABLE TTY KEYBOARD INTERRUPT
11422 043732 000207      RTS      PC          :: RETURN TO CALLER
11423
11424
11425      *TK SERVICE ROUTINE
11426      *THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
11427      *BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
11427      *IT IN THE QUEUE.

```



```

11428                                     ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
11429                                     ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)
11430
11431 043734 117746 135206 $TKSRV: MOVB  @STKB, -(SP)      ;; PICKUP THE CHARACTER
11432 043740 042716 177600      BIC    #↑C177, (SP)      ;; STRIP THE JUNK
11433 043744 021627 000003      CMP    (SP), #3         ;; IS IT A CONTROL C?
11434 043750 001007           BNE    1$              ;; BRANCH IF NO
11435 043752 104401 045050      TYPE   $CNTLC          ;; TYPE A CONTROL-C (↑C)
11436 043756 004737 043664      JSR    PC, $TKINT      ;; INIT THE KEYBOARD
11437 043762 005726           TST    (SP)+           ;; CLEAN UP STACK
11438 043764 000137 042442      JMP    CTRHLT          ;; CONTROL C RESTART
11439 043770 021627 000007      1$:  CMP    (SP), #7     ;; IS IT A CONTROL G?
11440 043774 001004           BNE    2$              ;; BRANCH IF NO
11441 043776 022737 000176 001140  CMP    #SWREG, SWR     ;; IS SOFT-SWR SELECTED?
11442 044004 001500           BEQ    6$              ;; GO TO SWR CHANGE
11443
11444 044006           2$:
11445 044006 022737 000001 043654  CMP    #1, $TKCNT     ;; IS THE QUEUE FULL?
11446 044014 001004           BNE    3$              ;; BRANCH IF NO
11447 044016 104401 001266      TYPE   $BELL           ;; RING THE TTY BELL
11448 044022 005726           TST    (SP)+           ;; CLEAN CHARACTER OFF OF STACK
11449 044024 000451           BR     5$              ;; EXIT
11450 044026 021627 000023      3$:  CMP    (SP), #23      ;; IS IT A CONTROL-S?
11451 044032 001021           BNE    32$             ;; BRANCH IF NO
11452 044034 005077 135104      CLR    @STKS           ;; DISABLE TTY KEYBOARD INTERRUPTS
11453 044040 005726           TST    (SP)+           ;; CLEAN CHAR OFF STACK
11454 044042 105777 135076      31$: TSTB   @STKS          ;; WAIT FOR A CHAR
11455 044046 100375           BPL    31$            ;; LOOP UNTIL ITS THERE
11456 044050 117746 135072      MOVB  @STKB, -(SP)    ;; GET THE CHARACTER
11457 044054 042716 177600      BIC    #↑C177, (SP)    ;; MAKE IT 7-BIT ASCII
11458 044060 022627 000021      CMP    (SP)+, #21     ;; IS IT A CONTROL-Q?
11459 044064 001366           BNE    31$            ;; BRANCH IF NO
11460 044066 012777 000100 135050  MOV    #100, @STKS    ;; REENABLE TTY KEYBOARD INTERRUPTS
11461 044074 000002           RTI                    ;; RETURN
11462 044076 005237 043654      32$: INC    $TKCNT        ;; COUNT THIS CHARACTER
11463 044102 021627 000140      CMP    (SP), #140     ;; IS IT UPPER CASE?
11464 044106 002405           BLT    4$              ;; BRANCH IF YES
11465 044110 021627 000175      CMP    (SP), #175     ;; IS IT A SPECIAL CHAR?
11466 044114 003002           BGT    4$              ;; BRANCH IF YES
11467 044116 042716 000040      BIC    #40, (SP)      ;; MAKE IT UPPER CASE
11468 044122 112677 177530      4$:  MOVB  (SP)+, @STKQIN  ;; AND PUT IT IN QUEUE
11469 044126 005237 043656      INC    $TKQIN         ;; UPDATE THE POINTER
11470 044132 023727 043656 043663  CMP    $TKQIN, #STKQEND ;; GO OFF THE END?
11471 044140 001003           BNE    5$              ;; BRANCH IF NO
11472 044142 012737 043662 043656  MOV    #STKQSR, $TKQIN ;; RESET THE POINTER
11473 044150 000002           5$:  RTI                    ;; RETURN
11474
11475                                     ;*****
11476                                     ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11477                                     ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
11478                                     ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11479                                     ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11480 044152 022737 000176 001140 $CKSWR: CMP    #SWREG, SWR  ;; IS THE SOFT-SWR SELECTED
11481 044160 001124           BNE    15$            ;; EXIT IF NOT
11482 044162 105777 134756      TSTB   @STKS          ;; IS A CHAR WAITING?
11483 044166 100121           BPL    15$            ;; IF NOT, EXIT

```


E01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.F11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 211
TTY INPUT ROUTINE

SEQ 0211

11484	044170	117746	134752		MOVB	@STKB, -(SP)	:: YES
11485	044174	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
11486	044200	021627	000007		CMP	(SP), #7	:: IS IT A CONTROL-G?
11487	044204	001300			BNE	2\$:: IF NOT, PUT IT IN THE TTY QUEUE
11488							:: AND EXIT
11489							
11490							
11491							
11492							
11493							
11494	044206	123727	001134	000001	6\$:	CMPB \$AUTOB, #1	:: ARE WE RUNNING IN AUTO-MODE?
11495	044214	001674				BEG 2\$:: BRANCH IF YES
11496	044216	005726				TST (SP)+	:: CLEAR CONTROL-G OFF STACK
11497	044220	004737	043664			JSR PC, \$TKINT	:: FLUSH THE TTY INPUT QUEUE
11498	044224	005077	134714			CLR @STKS	:: DISABLE TTY KEYBOARD INTERRUPTS
11499	044230	112737	000001	001135		MOVB #1, \$INTAG	:: SET INTERRUPT MODE INDICATOR
11500							
11501	044236	104401	045062			TYPE , \$CNTLG	:: ECHO THE CONTROL-G (↑G)
11502	044242	104401	045067		\$GTSWR:	TYPE , \$MSWR	:: TYPE CURRENT CONTENTS
11503	044246	013746	000176			MOV \$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
11504	044252	104402				TYPOC	:: GO TYPE--OCTAL ASCII(ALL DIGITS)
11505	044254	104401	045100			TYPE , \$MNEW	:: PROMPT FOR NEW SWR
11506	044260	005046			19\$:	CLR -(SP)	:: CLEAR COUNTER
11507	044262	005046				CLR -(SP)	:: THE NEW SWR
11508	044264	105777	134654		7\$:	TSTB @STKS	:: CHAR THERE?
11509	044270	100375				BPL 7\$:: IF NOT TRY AGAIN
11510							
11511	044272	117746	134650			MOVB @STKB, -(SP)	:: PICK UP CHAR
11512	044276	042716	177600			BIC #1C177, (SP)	:: MAKE IT 7-BIT ASCII
11513							
11514	044302	021627	000003			CMP (SP), #3	:: IS IT A CONTROL-C?
11515	044306	001015				BNE 9\$:: BRANCH IF NOT
11516	044310	104401	045050			TYPE , \$CNTLC	:: YES, ECHO CONTROL-C (↑C)
11517	044314	062706	000006			ADD #6, SP	:: CLEAN UP STACK
11518	044320	123727	001135	000001		CMPB \$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
11519	044326	001003				BNE 8\$:: BRANCH IF NO
11520	044330	012777	000100	134606		MOV #100, @STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
11521	044336	000137	042442		8\$:	JMP CTRHLT	:: CONTROL-C RESTART
11522							
11523							
11524	044342	021627	000025		9\$:	CMP (SP), #25	:: IS IT A CONTROL-U?
11525	044346	001005				BNE 10\$:: BRANCH IF NOT
11526	044350	104401	045055			TYPE , \$CNTLU	:: YES, ECHO CONTROL-U (↑U)
11527	044354	062706	000006		20\$:	ADD #6, SP	:: IGNORE PREVIOUS INPUT
11528	044360	000737				BR 19\$:: LET'S TRY IT AGAIN
11529							
11530							
11531	044362	021627	000015		10\$:	CMP (SP), #15	:: IS IT A <CR>?
11532	044366	001022				BNE 15\$:: BRANCH IF NO
11533	044370	005766	000004			TST 4(SP)	:: YES, IS IT THE FIRST CHAR?
11534	044374	001403				BEG 11\$:: BRANCH IF YES
11535	044376	016677	000002	134534		MOV 2(SP), @SWR	:: SAVE NEW SWR
11536	044404	062706	000006		11\$:	ADD #6, SP	:: CLEAN UP STACK
11537	044410	104401	001273		14\$:	TYPE , \$CRLF	:: ECHO <CR> AND <LF>
11538	044414	123727	001135	000001		CMPB \$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
11539	044422	001003				BNE 15\$:: BRANCH IF NOT

F01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZREKC.P11 01-OCT-76 13:09

MACY11 27(1006) 05-OCT-76 09:17 PAGE 212
TTY INPUT ROUTINE

SEQ 0212

```

11540 044424 012777 000100 134512      MOV      #100, @STKS      ;; RE-ENABLE TTY KBD INTERRUPTS
11541 044432 000002      15$: RTI                ;; RETURN
11542 044434 004737 043132      16$: JSR      PC, $TYPEC  ;; ECHO CHAR
11543 044440 021627 000060      CMP      (SP), #60      ;; CHAR < 0?
11544 044444 002420      BLT      18$           ;; BRANCH IF YES
11545 044446 021627 000067      CMP      (SP), #67      ;; CHAR > 7?
11546 044452 003015      BGT      18$           ;; BRANCH IF YES
11547 044454 042726 000060      BIC      #60, (SP)+     ;; STRIP-OFF ASCII
11548 044460 005766 000002      TST      2(SP)         ;; IS THIS THE FIRST CHAR
11549 044464 001403      BEQ      17$           ;; BRANCH IF YES
11550 044466 006316      ASL      (SP)         ;; NO, SHIFT PRESENT
11551 044470 006316      ASL      (SP)         ;; CHAR OVER TO MAKE
11552 044472 006316      ASL      (SP)         ;; ROOM FOR NEW ONE.
11553 044474 005266 000002      17$: INC      2(SP)     ;; KEEP COUNT OF CHAR
11554 044500 056616 177776      BIS      -2(SP), (SP)  ;; SET IN NEW CHAR
11555 044504 000667      BR       7$           ;; GET THE NEXT ONE
11556 044506 104401 001272      18$: TYPE     $QUES     ;; TYPE ?<CR><LF>
11557 044512 000720      BR       20$         ;; SIMULATE CONTROL-U
11558      .DSABL  LSB
11559
11560
11561      ;; *****
11562      ;; THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
11563      ;; *CALL:
11564      ;; * RDCHR                ;; GET A CHARACTER FROM THE QUEUE
11565      ;; * RETURN HERE         ;; CHARACTER IS ON THE STACK
11566      ;; *                     ;; WITH PARITY BIT STRIPPED OFF
11567      ;;
11568
11569 044514 011646      $RDCHR: MOV      (SP), -(SP)  ;; PUSH DOWN THE PC AND
11570 044516 016666 000004 000002  MOV      4(SP), 2(SP)  ;; THE PS
11571 044524 005066 000004      CLR      4(SP)        ;; GET READY FOR A CHARACTER
11572 044530 005046      CLR      -(SP)       ;; PUT NEW PS ON STACK
11573 044532 012746 044540      MOV      #64$, -(SP)  ;; PUT NEW PC ON STACK
11574 044536 000002      RTI                ;; POP NEW PC AND PS
11575 044540      64$:
11576 044540 005737 043654      1$: TST      $STKCNT   ;; WAIT ON A CHARACTER
11577 044544 001775      BEQ      1$
11578 044546 005337 043654      DEC      $STKCNT     ;; DECREMENT THE COUNTER
11579 044552 117766 177102 000004  MOVB     @STKQOUT, 4(SP) ;; GET ONE CHARACTER
11580 044560 005237 043660      INC      $STKQOUT    ;; UPDATE THE POINTER
11581 044564 023727 043660 043663  CMP      $STKQOUT, $STKGEND ;; DID IT GO OFF OF THE END?
11582 044572 001003      BNE     2$           ;; BRANCH IF NO
11583 044574 012737 043662 043660  MOV      #STKQSRT, $STKQOUT ;; RESET THE POINTER
11584 044602 000002      2$: RTI                ;; RETURN
11585      ;; *****
11586      ;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY
11587      ;; *CALL:
11588      ;; * RDLIN                ;; INPUT A STRING FROM THE TTY
11589      ;; * RETURN HERE         ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11590      ;; *                     ;; TERMINATOR WILL BE A BYTE OF ALL 0'S
11591
11592 044604 010346      $RDLIN: MOV      R3, -(SP)  ;; SAVE R3
11593 044606 005046      CLR      -(SP)       ;; CLEAR THE RUBOUT KEY
11594 044610 012703 045040      1$: MOV      $TTYIN, R3  ;; GET ADDRESS
11595 044614 022703 045050      2$: CMP      $TTYIN+8., R3 ;; BUFFER FULL?

```


11596	044620	101456			BLOS	4\$::BR IF YES
11597	044622	104410			RDCHR		::GO READ ONE CHARACTER FROM THE TTY
11598	044624	112613			MOVB	(SP)+,(R3)	::GET CHARACTER
11599	044626	122713	000177	10\$:	CMPB	#177,(R3)	::IS IT A RUBOUT
11600	044632	001022			BNE	5\$::BR IF NO
11601	044634	005716			TST	(SP)	::IS THIS THE FIRST RUBOUT?
11602	044636	001007			BNE	6\$::BR IF NO
11603	044640	112737	000134	045036	MOVB	#'\,9\$::TYPE A BACK SLASH
11604	044646	104401	045036		TYPE	9\$	
11605	044652	012716	177777		MOV	#-1,(SP)	::SET THE RUBOUT KEY
11606	044656	005303		6\$:	DEC	R3	::BACKUP BY ONE
11607	044660	020327	045040		CMP	R3,#\$TTYIN	::STACK EMPTY?
11608	044664	103434			BLO	4\$::BR IF YES
11609	044666	111337	045036		MOVB	(R3),9\$::SETUP TO TYPEOUT THE DELETED CHAR.
11610	044672	104401	045036		TYPE	9\$::GO TYPE
11611	044676	000746			BR	2\$::GO READ ANOTHER CHAR.
11612	044700	005716		5\$:	TST	(SP)	::RUBOUT KEY SET?
11613	044702	001406			BEG	7\$::BR IF NO
11614	044704	112737	000134	045036	MOVB	#'\,9\$::TYPE A BACK SLASH
11615	044712	104401	045036		TYPE	9\$	
11616	044716	005016			CLR	(SP)	::CLEAR THE RUBOUT KEY
11617	044720	122713	000025	7\$:	CMPB	#25,(R3)	::IS CHARACTER A CTRL U?
11618	044724	001003			BNE	8\$::BR IF NO
11619	044726	104401	045055		TYPE	\$CNTLU	::TYPE A CONTROL "U"
11620	044732	000726			BR	1\$::GO START OVER
11621	044734	122713	000022	8\$:	CMPB	#22,(R3)	::IS CHARACTER A "↑R"?
11622	044740	001011			BNE	3\$::BRANCH IF NO
11623	044742	105013			CLRB	(R3)	::CLEAR THE CHARACTER
11624	044744	104401	001273		TYPE	\$CRLF	::TYPE A "CR" & "LF"
11625	044750	104401	045040		TYPE	\$TTYIN	::TYPE THE INPUT STRING
11626	044754	000717			BR	2\$::GO PICKUP ANOTHER CHARACTER
11627	044756	104401	001272	4\$:	TYPE	\$QUES	::TYPE A "?"
11628	044762	000712			BR	1\$::CLEAR THE BUFFER AND LOOP
11629	044764	111337	045036	3\$:	MOVB	(R3),9\$::ECHO THE CHARACTER
11630	044770	104401	045036		TYPE	9\$	
11631	044774	122723	000015		CMPB	#15,(R3)+	::CHECK FOR RETURN
11632	045000	001305			BNE	2\$::LOOP IF NOT RETURN
11633	045002	105063	177777		CLRB	-1(R3)	::CLEAR RETURN (THE 15)
11634	045006	104401	001274		TYPE	\$LF	::TYPE A LINE FEED
11635	045012	005726			TST	(SP)+	::CLEAN RUBOUT KEY FROM THE STACK
11636	045014	012603			MOV	(SP)+,R3	::RESTORE R3
11637	045016	011646			MOV	(SP)-,(SP)	::ADJUST THE STACK AND PUT ADDRESS OF THE
11638	045020	016666	000004	000002	MOV	4(SP),2(SP)	::FIRST ASCII CHARACTER ON IT
11639	045026	012766	045040	000004	MOV	#\$TTYIN,4(SP)	
11640	045034	000002			RTI		::RETURN
11641	045036	000		9\$:	.BYTE	0	::STORAGE FOR ASCII CHAR. TO TYPE
11642	045037	000			.BYTE	0	::TERMINATOR
11643	045040	000010			\$TTYIN:	.BLKB	8.
11644	045050	041536	005015	000	\$CNTLC:	.ASCIZ	/↑C/<15><12>
11645	045055	136	006525	000012	\$CNTLU:	.ASCIZ	/↑U/<15><12>
11646	045062	043536	005015	000	\$CNTLG:	.ASCIZ	/↑G/<15><12>
11647	045067	015	051412	051127	\$MSWR:	.ASCIZ	<15><12>/SWR = /
11648	045074	036440	000040				
11649	045100	020040	042516	020127	\$MNEW:	.ASCIZ	/ NEW = /
11650	045106	020075	000				
11651	045112				.EVEN		

H01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:09

MACY11 27(1006) 05-OCT-76 09:17 PAGE 214
READ AN OCTAL NUMBER FROM THE TTY

SEQ 0214

```
11652 .SBTTL READ AN OCTAL NUMBER FROM THE TTY
11653
11654
11655
11656
11657
11658
11659
11660
11661
11662
11663
11664
11665
11666 045112 011646
11667 045114 016666 000004 000002
11668 045122 010046
11669 045124 010146
11670 045126 010246
11671 045130 104411
11672 045132 012600
11673 045134 010037 045240
11674 045140 005001
11675 045142 005002
11676 045144 112046
11677 045146 001420
11678 045150 122716 000060
11679 045154 003026
11680 045156 122716 000067
11681 045162 002423
11682 045164 006301
11683 045166 006102
11684 045170 006301
11685 045172 006102
11686 045174 006301
11687 045176 006102
11688 045200 042716 177770
11689 045204 062601
11690 045206 000756
11691 045210 005726
11692 045212 010166 000012
11693 045216 010237 045250
11694 045222 012602
11695 045224 012601
11696 045226 012600
11697 045230 000002
11698 045232 005726
11699 045234 105010
11700 045236 104401
11701 045240 000000
11702 045242 104401 001272
11703 045246 000730
11704 045250 000000
11705
11706
11707

;*****
;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
;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
;CALL:
;* RDOCT ;:READ AN OCTAL NUMBER
;* RETURN HERE ;:LOW ORDER BITS ARE ON TOP OF THE STACK
;* ;:HIGH ORDER BITS ARE IN $HIOCT

$RDOCT: MOV (SP),-(SP) ;:PROVIDE SPACE FOR THE
MOV 4(SP),2(SP) ;:INPUT NUMBER
MOV RO,-(SP) ;:PUSH RO ON STACK
MOV R1,-(SP) ;:PUSH R1 ON STACK
MOV R2,-(SP) ;:PUSH R2 ON STACK
1$: RDLIN ;:READ AN ASCII LINE
MOV (SP)+,RO ;:GET ADDRESS OF 1ST CHARACTER
MOV RO,$$ ;:AND SAVE IT
CLR R1 ;:CLEAR DATA WORD
CLR R2
2$: MOVB (RO)+,-(SP) ;:PICKUP THIS CHARACTER
BEQ 3$ ;:IF ZERO GET OUT
CMPB #'0,(SP) ;:MAKE SURE THIS CHARACTER
BGT 4$ ;:IS AN OCTAL DIGIT
CMPB #'7,(SP)
BLT 4$
ASL R1 ;:*2
ROL R2 ;:*4
ASL R1 ;:*8
ROL R2 ;:*8
BIC #'C7,(SP) ;:STRIP THE ASCII JUNK
ADD (SP)+,R1 ;:ADD IN THIS DIGIT
BR 2$ ;:LOOP
3$: TST (SP)+ ;:CLEAN TERMINATOR FROM STACK
MOV R1,12(SP) ;:SAVE THE RESULT
MOV R2,$HIOCT
MOV (SP)+,R2 ;:POP STACK INTO R2
MOV (SP)+,R1 ;:POP STACK INTO R1
MOV (SP)+,RO ;:POP STACK INTO RO
RTI ;:RETURN
4$: TST (SP)+ ;:CLEAN PARTIAL FROM STACK
CLRB (RO) ;:SET A TERMINATOR
TYPE ;:TYPE UP THRU THE BAD CHAR.
5$: .WORD 0
TYPE $QUES ;: "?" "CR" & "LF"
BR 1$ ;:TRY AGAIN
$HIOCT: .WORD 0 ;:HIGH ORDER BITS GO HERE
.SBTTL SAVE AND RESTORE RO-R5 ROUTINES
;*****
```



```

11708
11709
11710
11711
11712
11713
11714
11715
11716
11717
11718
11719
11720
11721
11722 045252
11723 045252 010046
11724 045254 010146
11725 045256 010246
11726 045260 010346
11727 045262 010446
11728 045264 010546
11729 045266 016646 000022
11730 045272 016646 000022
11731 045276 016646 000022
11732 045302 016646 000022
11733 045306 000002
11734
11735
11736
11737
11738 045310
11739 045310 012666 000022
11740 045314 012666 000022
11741 045320 012666 000022
11742 045324 012666 000022
11743 045330 012605
11744 045332 012604
11745 045334 012603
11746 045336 012602
11747 045340 012601
11748 045342 012600
11749 045344 000002
11750
11751
11752
11753
11754 045346 012737 045506 000024
11755 045354 012737 000340 000026
11756 045362 010046
11757 045364 010146
11758 045366 010246
11759 045370 010346
11760 045372 010446
11761 045374 010546
11762 045376 017746 133536
11763 045402 010637 045512

```

```

;*SAVE RO-R5
;*CALL:
;* SAVREG
;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
;*
;*TOP----(+16)
;* +2----(+18)
;* +4----R5
;* +6----R4
;* +8----R3
;*+10----R2
;*+12----R1
;*+14----R0

$SAVREG:
MOV RO,-(SP) ;;PUSH RO 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 $ILLUP,@#PWRVEC ;;SET FOR FAST UP
MOV #340,@#PWRVEC+2 ;;PRIO:7
MOV RO,-(SP) ;;PUSH RO ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
MOV R4,-(SP) ;;PUSH R4 ON STACK
MOV R5,-(SP) ;;PUSH R5 ON STACK
MOV @SWR,-(SP) ;;PUSH @SWR ON STACK
MOV SP,$SAVR6 ;;SAVE SP

```



```

11764 045406 012737 045420 000024      MOV    #PWRUP,@PWRVEC ;;SET UP VECTOR
11765 045414 000000                      HALT
11766 045416 000776                      BR     .-2                ;;HANG UP
11767
11768
11769
11770 045420 012737 045506 000024      $PWRUP: MOV    #SILLUP,@PWRVEC ;;SET FOR FAST DOWN
11771 045426 013706 045512                      MOV    $SAVR6,SP        ;;GET SP
11772 045432 005037 045512                      CLR    $SAVR6           ;;WAIT LOOP FOR THE TTY
11773 045436 005237 045512      1$:    INC    $SAVR6           ;;WAIT FOR THE INC
11774 045442 001375                      BNE    1$              ;;OF WORD
11775 045444 012677 133470                      MOV    (SP)+,@SWR       ;;POP STACK INTO @SWR
11776 045450 012605                      MOV    (SP)+,R5        ;;POP STACK INTO R5
11777 045452 012604                      MOV    (SP)+,R4        ;;POP STACK INTO R4
11778 045454 012603                      MOV    (SP)+,R3        ;;POP STACK INTO R3
11779 045456 012602                      MOV    (SP)+,R2        ;;POP STACK INTO R2
11780 045460 012601                      MOV    (SP)+,R1        ;;POP STACK INTO R1
11781 045462 012600                      MOV    (SP)+,R0        ;;POP STACK INTO R0
11782 045464 012737 045346 000024      MOV    #PWRDN,@PWRVEC ;;SET UP THE POWER DOWN VECTOR
11783 045472 012737 000340 000026      MOV    #340,@PWRVEC+2 ;;PRIO:7
11784 045500 104401                      TYPE   $POWER           ;;REPORT THE POWER FAILURE
11785 045502 045514      $PWRMG: .WORD $POWER      ;;POWER FAIL MESSAGE POINTER
11786 045504 000002                      RTI
11787 045506 000000      $SILLUP: HALT           ;;THE POWER UP SEQUENCE WAS STARTED
11788 045510 000776                      BR     .-2                ;;BEFORE THE POWER DOWN WAS COMPLETE
11789 045512 000000      $SAVR6: 0                ;;PUT THE SP HERE
11790 045514 005015 047520 042527      $POWER: .ASCIZ <15><12>"POWER"
11791 045522 000122
11792
11793      .SBTTL .EVEN
11794      .SBTTL TRAP DECODER
11795
11796
11797
11798
11799
11800
11801 045524 010046      $TRAP:  MOV    RO,-(SP)      ;;SAVE RO
11802 045526 016600 000002      MOV    2(SP),RO          ;;GET TRAP ADDRESS
11803 045532 005740      TST    -(RO)            ;;BACKUP BY 2
11804 045534 111000      MOV    (RO),RO          ;;GET RIGHT BYTE OF TRAP
11805 045536 006300      ASL    RO                ;;POSITION FOR INDEXING
11806 045540 016000 045560      MOV    $TRPAD(RO),RO    ;;INDEX TO TABLE
11807 045544 000200      RTS    RO                ;;GO TO ROUTINE
11808
11809
11810      ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
11811
11812 045546 011646      $TRAP2: MOV    (SP),-(SP)  ;;MOVE THE PC DOWN
11813 045550 016666 000004 000002      MOV    4(SP),2(SP)     ;;MOVE THE PSW DOWN
11814 045556 000002      RTI                      ;;RESTORE THE PSW
11815
11816      .SBTTL TRAP TABLE
11817
11818      ;;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11819      ;;BY THE "TRAP" INSTRUCTION.

```


Address	Word	Byte	Label	Call	Description
11820					
11821					
11822					
11823	045560	045546	\$TRPAD:	WORD	\$TRAP2
11824	045562	042720		\$TYPE	::CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
11825	045564	043452		\$TYPOC	::CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11826	045566	043426		\$TYPOS	::CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
11827	045570	043466		\$TYPON	::CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
11828	045572	043202		\$TYPDS	::CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
11829					
11830	045574	044242		\$GTSWR	::CALL=GTSWR TRAP+6(104406) GET SOFT-SWR SETTING
11831					
11832	045576	044152		\$CKSWR	::CALL=CKSWR TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
11833	045600	044514		\$RDCHR	::CALL=RDCHR TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
11834	045602	044604		\$RDLIN	::CALL=RDLIN TRAP+11(104411) TTY TYPEIN STRING ROUTINE
11835	045604	045112		\$RDOCT	::CALL=RDOCT TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
11836	045606	045252		\$SAVREG	::CALL=SAVREG TRAP+13(104413) SAVE RD-R5 ROUTINE
11837	045610	045310		\$RESREG	::CALL=RESREG TRAP+14(104414) RESTORE RD-R5 ROUTINE
11838	045612	034122		\$SCOPI	::CALL=SCOPI TRAP+15(104415) INTERNAL LOOP ON ERROR
11839	045614	035722		\$SSINIT	::CALL=SSINIT TRAP+16(104416) INITIALIZE SUBSYSTEM
11840	045616	034634		\$LOADRK	::CALL=TLOADRK TRAP+17(104417) LOAD RK611 FOR OPERATION
11841	045620	034716		\$GETRK	::CALL=TGETRK TRAP+20(104420) GET RK611 REGISTERS
11842	045622	036706		\$CHKOP	::CALL=TCHKOP TRAP+21(104421) CHECK OPERATION FOR ANY ERRORS
11843	045624	036654		\$CHKWE	::CALL=TCHKWE TRAP+22(104422) CHECK OPERATION FOR EXPECTED ERRORS
11844	045626	034430		\$IWAT16	::CALL=TWAT16 TRAP+23(104423) WAIT 16 MS
11845	045630	034420		\$IWAT32	::CALL=TWAT32 TRAP+24(104424) WAIT 32 MS
11846	045632	034410		\$IWAT48	::CALL=TWAT48 TRAP+25(104425) WAIT 48 MS
11847	045634	034400		\$IWAT64	::CALL=TWAT64 TRAP+26(104426) WAIT 64 MS
11848	045636	034370		\$IWAT80	::CALL=TWAT80 TRAP+27(104427) WAIT 80 MS
11849	045640	034360		\$IWAT96	::CALL=TWAT96 TRAP+30(104430) WAIT 96 MS
11850	045642	034350		\$IWAT112	::CALL=TWAT112 TRAP+31(104431) WAIT 112 MS
11851	045644	034340		\$IWAT128	::CALL=TWAT128 TRAP+32(104432) WAIT 128 MS
11852	045646	034330		\$IWAT144	::CALL=TWAT144 TRAP+33(104433) WAIT 144 MS
11853	045650	034320		\$IWAT159	::CALL=TWAT159 TRAP+34(104434) WAIT 160 MS
11854	045652	034310		\$IWAT15	::CALL=TWAT15 TRAP+35(104435) WAIT FOR 1 SECOND
11855	045654	034300		\$IWAT25	::CALL=TWAT25 TRAP+36(104436) WAIT FOR 2 SECONDS
11856	045656	034260		\$IWAT85	::CALL=TWAT85 TRAP+37(104437) WAIT FOR 8 SECONDS
11857	045660	034270		\$IWAT1M	::CALL=TWAT1M TRAP+40(104440) WAIT FOR 1 MIN
11858		000102			

\$TERM=-\$TRPAD


```

11859
11860
11861
11862
11863
11864
11865 045662
11866 045662 177777
11867 045664 177777
11868 045666 177777
11869 045670 052525
11870 045672 052525
11871 045674 052525
11872 045676 177777
11873 045700 177777
11874 045702 052525
11875 045704 052525
11876 045706 177777
11877 045710 052525
11878 045712 177252
11879 045714 177252
11880 045716 172765
11881 045720 172765
11882
11883
11884
11885 045722
11886 045722 000000
11887 045724 000000
11888 045726 000000
11889 045730 177777
11890 045732 177777
11891 045734 177777
11892 045736 000000
11893 045740 000000
11894 045742 177777
11895 045744 177777
11896 045746 000000
11897 045750 177777
11898 045752 000000
11899 045754 177777
11900 045756 000000
11901 045760 177777
11902
11903
11904
11905 045762
11906 045762 052525
11907 045764 052525
11908 045766 052525
11909 045770 125252
11910 045772 125252
11911 045774 125252
11912 045776 052525
11913 046000 052525
11914 046002 125252

```

```

.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
177777
000000
177777
000000
177777
000000
177777

```

```

;DATA PATTERN 4
; LO FREQ. PHASE MIX
PAT04:

```

```

052525
052525
052525
125252
125252
125252
052525
052525
125252

```


11915	046004	125252
11916	046006	052525
11917	046010	125252
11918	046012	052525
11919	046014	125252
11920	046016	052525
11921	046020	125252

125252
052525
125252
052525
125252
052525
125252

```

:DATA PATTERN 5
:
:MAX PRECOMP. PHASE MIX

```

PAT05:

11925	046022	
11926	046022	133333
11927	046024	066666
11928	046026	155555
11929	046030	155555
11930	046032	133333
11931	046034	066666
11932	046036	066666
11933	046040	155555
11934	046042	155555
11935	046044	133333
11936	046046	133333
11937	046050	133333
11938	046052	133333
11939	046054	133333
11940	046056	133333
11941	046060	133333

133333
066666
155555
155555
133333
066666
066666
155555
155555
133333
133333
133333
133333
133333
133333
133333
133333
133333

```

:DATA PATTERN 6
:
:ROTATING BOUNDARY PULSE PRECOMP.

```

PAT06:

11945	046062	
11946	046062	121105
11947	046064	150442
11948	046066	064221
11949	046070	132110
11950	046072	055044
11951	046074	026422
11952	046076	013211
11953	046100	105504
11954	046102	042642
11955	046104	021321
11956	046106	110550
11957	046110	044264
11958	046112	022132
11959	046114	011055
11960	046116	104426
11961	046120	042213

121105
150442
064221
132110
055044
026422
013211
105504
042642
021321
110550
044264
022132
011055
104426
042213

```

:DATA PATTERN 7
:
:FIELD OF ALL ONES

```

```

:DATA PATTERN 10
:
:ROTATING CELL PULSE PRECOMP.

```

PAT10:

11968	046122	
11969	046122	026455
11970	046124	113226

026455
113226

11971	046126	045513	045513
11972	046130	122645	122645
11973	046132	151322	151322
11974	046134	064551	064551
11975	046136	132264	132264
11976	046140	055132	055132
11977	046142	026455	026455
11978	046144	113226	113226
11979	046146	045513	045513
11980	046150	122645	122645
11981	046152	151322	151322
11982	046154	064551	064551
11983	046156	132264	132264
11984	046160	055132	055132

;DATA PATTERN 11
SHIFTED 1 IN A FIELD OF ZEROS

11985			
11986			
11987			
11988	046162		
11989	046162	000001	000001
11990	046164	000002	000002
11991	046166	000004	000004
11992	046170	000010	000010
11993	046172	000020	000020
11994	046174	000040	000040
11995	046176	000100	000100
11996	046200	000200	000200
11997	046202	000400	000400
11998	046204	001000	001000
11999	046206	002000	002000
12000	046210	004000	004000
12001	046212	010000	010000
12002	046214	020000	020000
12003	046216	040000	040000
12004	046220	100000	100000

;DATA PATTERN 12
SHIFTED 0 IN A FIELD OF ONES

12005			
12006			
12007			
12008	046222		
12009	046222	177776	177776
12010	046224	177775	177775
12011	046226	177773	177773
12012	046230	177767	177767
12013	046232	177757	177757
12014	046234	177737	177737
12015	046236	177677	177677
12016	046240	177577	177577
12017	046242	177377	177377
12018	046244	176777	176777
12019	046246	175777	175777
12020	046250	173777	173777
12021	046252	167777	167777
12022	046254	157777	157777
12023	046256	137777	137777
12024	046260	077777	077777

;DATA PATTERN 13

12025
12026

12027		
12028	046262	
12029	046262	052525
12030	046264	052525
12031	046266	052525
12032	046270	052525
12033	046272	052525
12034	046274	052525
12035	046276	052525
12036	046300	052525
12037	046302	052525
12038	046304	052525
12039	046306	052525
12040	046310	052525
12041	046312	052525
12042	046314	052525
12043	046316	052525
12044	046320	052525

PAT13: ALTERNATING 0-1

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

:DATA PATTERN 14
PAT14: ALTERNATING 1-0

12047		
12048	046322	
12049	046322	125252
12050	046324	125252
12051	046326	125252
12052	046330	125252
12053	046332	125252
12054	046334	125252
12055	046336	125252
12056	046340	125252
12057	046342	125252
12058	046344	125252
12059	046346	125252
12060	046350	125252
12061	046352	125252
12062	046354	125252
12063	046356	125252
12064	046360	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

12067		
12068	046362	
12069	046362	000001
12070	046364	000003
12071	046366	000007
12072	046370	000017
12073	046372	000037
12074	046374	000077
12075	046376	000177
12076	046400	000377
12077	046402	000777
12078	046404	001777
12079	046406	003777
12080	046410	007777
12081	046412	017777
12082	046414	037777

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

12083	046416	077777
12084	046420	177777
12085		
12086		
12087		
12088	0464	
12089	0464	072307
12090	0464	135143
12091	0464	156461
12092	0464	167230
12093	0464	167230
12094	0464	073514
12095	0464	035646
12096	0464	035646
12097	0464	016723
12098	0464	107351
12099	0464	143564
12100	0464	061672
12101	0464	030735
12102	0464	114356
12103	0464	046167
12104	0464	123073
12105	0464	151453
12106	0464	164616

:DATA PATTERN 16
: PAT16: COMPOSITE ROTATING

077777
177777
072307
135143
156461
167230
167230
073514
035646
035646
016723
107351
143564
061672
030735
114356
046167
123073
151453
164616

12106			.SBTTL	FIELDS AND VARIABLES FOR OPERATION CHECKING	
12107	024000		CS1ERBIT=24000		:CS1 ERROR BITS SPAR & CTO
12108	177400		CS2ERBIT=177400		:CS2 ERROR BITS
12109					:DLT,WCE,UPE,NED,NEM
12110					:PGE,MOS,UFE
12111	000070		DSERBIT=70		:DS ERROR BITS
12112					:SPDLSS,DROT,ACLO
12113					
12114	046462	000000	EXPWC: .WORD	0	:EXPECTED WORD COUNT (GIVEN)
12115	046464	000000	EXPLUBA: .WORD	0	:EXPECTED UPPER BA (COMPUTED)
12116	046466	000000	EXPBA: .WORD	0	:EXPECTED BUS ADDRESS (COMPUTED)
12117	046470	000000	EXPCYL: .WORD	0	:EXPECTED CYLINDER (COMPUTED)
12118	046472	000000	EXPSEC: .WORD	0	:EXPECTED SECTOR (COMPUTED)
12119	046474	000000	EXPTRK: .WORD	0	:EXPECTED TRACK (COMPUTED)
12120					
12121	046476	000000	REALWC: .WORD	0	:WORD COUNT AT END OF OPERATION
12122	046500	000000	REALUB: .WORD	0	:REAL UPPER BA
12123	046502	000000	REALBA: .WORD	0	:BUS ADDRESS
12124	046504	000000	REALCY: .WORD	0	:CYLINDER
12125	046506	000000	REALTRK: .WORD	0	:TRACK
12126	046510	000000	REALSEC: .WORD	0	:SECTOR
12127					
12128	046512	000000	GRP1ER: .WORD	0	:GROUP 1 ERROR FIELDS
12129		000001	SPARERR=BIT0		:CONTROLLER DETECTED DRIVE BUS PARITY ERR
12130		000002	SKIERR= BIT1		:SEEK INCOMPLETE
12131		000004	NXFERR= BIT2		:NON-EXECUTABLE DRIVE FUNCTION
12132		000010	DRPARERR=BIT3		:DRIVE DETECTED DRIVE BUS PARITY ERROR
12133		000020	FMTERR= BIT4		:FORMAT ERROR
12134		000040	DTYERR= BITS		:DRIVE TYPE ERROR
12135		000100	ACLOERR=BIT6		:AC LOW ERROR
12136		000200	SPDERR= BIT7		:SPEED LOSS ERROR
12137		000400	DROTERR=BIT8		:DRIVE OFF TRACK ERROR
12138		001000	COERR= BIT9		:CYLINDER OVER FLOW ERROR
12139		002000	IDAERR= BIT10		:ILLEGAL DISK ADDRESS ERROR
12140		004000	WLERR= BIT11		:WRITE LOCK ERROR
12141		010000	DTERR= BIT12		:DRIVE TIMING ERROR
12142		020000	NCERWE= BIT13		:NO CERR WITH ERROR SET ERROR
12143		040000	UNSERR= BIT14		:DRIVE UNSAFE ERROR
12144		100000	CERNER= BIT15		:CERR BUT NO ERROR SET ERROR
12145					
12146	046514	000000	GRP2ER: .WORD	0	:GROUP 2 ERROR FIELD
12147		000001	ECHERR= BIT0		:ECC HARD ERROR
12148		000002	DCKERR= BIT1		:DATA CHECK ERROR
12149		000004	WCKERR= BIT2		:WRITE CHECK ERROR
12150		000010	DLTERR= BIT3		:DATA LATE ERROR
12151		000020	OPIERR= BIT4		:OPERATION INCOMPLETE ERROR
12152		000040	HVRCERR=BIT5		:HEADER VRC ERROR
12153		000100	BSERR= BIT6		:BAD SECTOR ERROR
12154					
12155	046516	000000	GRP3ER: .WORD	0	:GROUP 3 ERROR FIELD
12156		000001	NEDERR= BIT0		:NON-EXISTANT DRIVE ERROR
12157		000002	CTOERR= BIT1		:CONTROLLER TIME OUT ERROR
12158		000004	UFERR= BIT2		:UNIT FIELD ERROR
12159		000010	MDSERR= BIT3		:MULTIPLE DRIVE SELECT ERROR
12160		000020	PGERR= BIT4		:PROGRAMMING ERROR
12161		000040	NEMERR= BITS		:NON-EXISTANT MEMORY ERROR

E02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.F11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 224
FIELDS AND VARIABLES FOR OPERATION CHECKING

SEQ 0224

12162		000100	UPERR= BIT6	:UNIBUS PARITY ERROR
12163		000200	ILFERR= BIT7	:ILLEGAL FUNCTION ERROR.
12164				
12165	046520	000000	GRP4ER: .WORD 0	:GROUP 4 ERROR FIELD
12166		000001	WCERR= BIT0	:WORD COUNT ERROR FLAG
12167		000002	UBAERR= BIT1	:UPPER BA ERROR
12168		000004	BAERR= BIT2	:BUS ADDRESS ERROR FLAG
12169		000010	CYLERR= BIT3	:CYL ADDRESS ERROR FLAG
12170		000020	TRKERR= BIT4	:TRACK ADDRESS ERROR FLAG
12171		000040	SECERR= BITS	:SECTOR ADDRESS ERROR FLAG
12172				
12173	046522	052707	GRP4MS: .WORD EM10	
12174	046524	052762	.WORD EM11A	
12175	046526	052734	.WORD EM11	
12176	046530	053036	.WORD EM12	
12177	046532	053071	.WORD EM13	
12178	046534	053121	.WORD EM14	
12179				
12180	046536	053153	GRP3MS: .WORD EM15	
12181	046540	053176	.WORD EM16	
12182	046542	053221	.WORD EM17	
12183	046544	053242	.WORD EM18	
12184	046546	053267	.WORD EM19	
12185	046550	053311	.WORD EM20	
12186	046552	053335	.WORD EM21	
12187	046554	053361	.WORD EM22	
12188				
12189	046556	053402	GRP2MS: .WORD EM23	
12190	046560	053413	.WORD EM24	
12191	046562	053426	.WORD EM25	
12192	046564	053442	.WORD EM26	
12193	046566	053454	.WORD EM27	
12194	046570	053501	.WORD EM28	
12195	046572	053514	.WORD EM29	
12196				
12197	046574	053535	GRP1MS: .WORD EM30	
12198	046576	053610	.WORD EM31	
12199	046600	053630	.WORD EM32	
12200	046602	053666	.WORD EM33	
12201	046604	053734	.WORD EM34	
12202	046606	053751	.WORD EM35	
12203	046610	053772	.WORD EM36	
12204	046612	054001	.WORD EM37	
12205	046614	054024	.WORD EM38	
12206	046616	054044	.WORD EM39	
12207	046620	054066	.WORD EM40	
12208	046622	054120	.WORD EM41	
12209	046624	054141	.WORD EM42	
12210	046626	054164	.WORD EM43	
12211	046630	054226	.WORD EM44	
12212	046632	054243	.WORD EM45	
12213				
12214	046634	000000	GPSUMF: .WORD 0	:GROUP ERROR SUMMARY FLAGS
12215		000001	GRP1ST= BIT0	:GROUP 1 ERROR SET
12216		000002	GRP2ST= BIT1	:GROUP 2 ERROR SET
12217		000004	GRP3ST= BIT2	:GROUP 3 ERROR SET

12218	000010	GP1NR=	BIT3	:GROUP 1 ERROR NOT RECEIVED
12219	000020	GP2NR=	BIT4	:GROUP 2 ERROR NOT RECEIVED
12220	000040	GP3NR=	BIT5	:GROUP 3 ERROR NOT RECEIVED
12221	040000	DRSTER=	BIT14	:ERROR IN GETTING DRIVE STATUS FLAG.
12222	100000	REPNR=	BIT15	:REPORTING NOT RECEIVED SWITCH

12223
12224 .SBTTL TABLE OF OPERATION MESSAGE ADDRESS
12225 :* THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE
12226 :* USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.
12227

12228	046636	046676	CMNDLB: .WORD	OPER00	:ADDRESS OF SELECT MESSAGE
12229	046640	046713	.WORD	OPER02	:PACK ACK
12230	046642	046724	.WORD	OPER04	:DRIVE CLEAR
12231	046644	046740	.WORD	OPER06	:UNLOAD
12232	046646	046747	.WORD	OPER10	:START SPINDLE
12233	046650	046765	.WORD	OPER12	:RECALIBRATE
12234	046652	047001	.WORD	OPER14	:OFFSET
12235	046654	047010	.WORD	CPER16	:SEEK
12236	046656	047015	.WORD	OPER20	:READ DATA
12237	046660	047027	.WORD	OPER22	:WRITE DATA
12238	046662	047042	.WORD	OPER24	:READ HEADER
12239	046664	047057	.WORD	OPER26	:WRITE HEADER
12240	046666	047075	.WORD	OPER30	:WRITE CHECK
12241	046670	047111	.WORD	OPER32	:ILLEGAL OPERATION 33
12242	046672	047135	.WORD	OPER34	:35
12243	046674	047161	.WORD	OPER36	:37
12244					

12245 .SBTTL OPERATION MESSAGES
12246 046676 051104 053111 020105 OPER00: .ASCIZ /DRIVE SELECT/
12247 046704 042523 042514 052103
12248 046712 000
12249 046713 120 041501 020113 OPER02: .ASCIZ /PACK ACK/
12250 046720 041501 000113
12251 046724 051104 053111 020105 OPER04: .ASCIZ /DRIVE CLEAR/
12252 046732 046103 040505 000122
12253 046740 047125 047514 042101 OPER06: .ASCIZ /UNLOAD/
12254 046746 000
12255 046747 123 040524 052122 OPER10: .ASCIZ /START SPINDLE/
12256 046754 051440 044520 042116
12257 046762 042514 000
12258 046765 122 041505 046101 OPER12: .ASCIZ /RECALIBRATE/
12259 046772 041111 040522 042524
12260 047000 000
12261 047001 117 043106 042523 OPER14: .ASCIZ /OFFSET/
12262 047006 000124
12263 047010 042523 045505 000 OPER16: .ASCIZ /SEEK/
12264 047015 122 040505 020104 OPER20: .ASCIZ /READ DATA/
12265 047022 040504 040524 000
12266 047027 127 044522 042524 OPER22: .ASCIZ /WRITE DATA/
12267 047034 042040 052101 000101
12268 047042 042522 042101 044040 OPER24: .ASCIZ /READ HEADERS/
12269 047050 040505 042504 051522
12270 047056 000
12271 047057 127 044522 042524 OPER26: .ASCIZ /WRITE HEADERS/
12272 047064 044040 040505 042504
12273 047072 051522 000

12274	047075	127	044522	042524	OPER30: .ASCIZ /WRITE CHECK/
12275	047102	041440	042510	045503	
12276	047110	000			
12277	047111	111	046114	043505	OPER32: .ASCIZ /ILLEGAL FUNCTION 33/
12278	047116	046101	043040	047125	
12279	047124	052103	047511	020116	
12280	047132	031463	000		
12281	047135	111	046114	043505	OPER34: .ASCIZ /ILLEGAL FUNCTION 35/
12282	047142	046101	043040	047125	
12283	047150	052103	047511	020116	
12284	047156	032463	000		
12285	047161	111	046114	043505	OPER36: .ASCIZ /ILLEGAL FUNCTION 37/
12286	047166	046101	043040	047125	
12287	047174	052103	047511	020116	
12288	047202	033463	000		
12289	047205	127	044522	042524	OPER37: .ASCIZ /WRITE DATA ABORTED WITH BSE/
12290	047212	042040	052101	020101	
12291	047220	041101	051117	042524	
12292	047226	020104	044527	044124	
12293	047234	041040	042523	000	
12294	047241				OPER40:
12295	047241	127	044522	042524	OPER41: .ASCIZ /WRITE CHECK ABORTED WITH WCE/
12296	047246	041440	042510	045503	
12297	047254	040440	047502	052122	
12298	047262	042105	053440	052111	
12299	047270	020110	041527	000105	
12300	047276	051127	052111	020105	OPER42: .ASCIZ /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
12301	047304	040504	040524	040440	
12302	047312	047502	052122	042105	
12303	047320	053440	052111	020110	
12304	047326	047516	026516	054105	
12305	047334	051511	040524	052116	
12306	047342	046440	046505	051117	
12307	047350	000131			
12308	047352	042522	042101	042040	OPER43: .ASCIZ /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
12309	047360	052101	020101	041101	
12310	047366	051117	042524	020104	
12311	047374	044527	044124	047040	
12312	047402	047117	042455	044530	
12313	047410	052123	047101	020124	
12314	047416	042515	047515	054522	
12315	047424	000			
12316	047425	127	044522	042524	OPER44: .ASCIZ /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR/
12317	047432	042040	052101	020101	
12318	047440	041101	051117	042524	
12319	047446	020104	044527	044124	
12320	047454	052440	044516	052502	
12321	047462	020123	040520	044522	
12322	047470	054524	042440	051122	
12323	047476	051117	000		
12324					

12325				
12326				
12327	047501	040	000040	
12328	047504	005015	045522	030466
12329	047512	020061	052502	020123
12330	047520	042101	051104	051505
12331	047526	020123	020050	000
12332	047533	040	020051	020075
12333	047540	000		
12334	047541	122	033113	030461
12335	047546	053040	041505	047524
12336	047554	020122	042101	051104
12337	047562	051505	020123	020050
12338	047570	000		
12339	047571	122	033113	030461
12340	047576	050040	044522	051117
12341	047604	052111	020131	020050
12342	047612	000		
12343	047613	111	051516	043125
12344	047620	044506	044503	047105
12345	047626	020124	042515	047515
12346	047634	054522	020056	051120
12347	047642	043517	040522	020115
12348	047650	041101	051117	044524
12349	047656	043516	006456	000012
12350	047664	005015	047524	041040
12351	047672	050131	051501	020123
12352	047700	042524	052123	047111
12353	047706	020107	051104	053111
12354	047714	020105	026060	050040
12355	047722	040514	042503	044440
12356	047730	020124	043117	026506
12357	047736	044514	042516	
12358	047742	005015	047524	052040
12359	047750	051505	020124	051104
12360	047756	053111	020105	026060
12361	047764	051040	050105	040514
12362	047772	042503	050040	047522
12363	050000	051107	046501	050040
12364	050006	041501	020113	044527
12365	050014	044124	051440	051103
12366	050022	052101	044103	050040
12367	050030	041501	000113	
12368	050034	005015	051104	053111
12369	050042	020105	020060	044527
12370	050050	046114	047040	052117
12371	050056	041040	020105	042524
12372	050064	052123	042105	020056
12373	050072	052123	051101	020124
12374	050100	052101	031040	032061
12375	050106	052040	020117	042524
12376	050114	052123	042040	020122
12377	050122	000061		
12378	050124	005015	047516	042040
12379	050132	044522	042526	020123
12380	050140	053101	044501	040514

.SBTTL ASCII MESSAGES

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: .ASCIZ <15><12>/DRIVE 0 WILL NOT BE TESTED. START AT 214 TO TEST DR 1/

OPR008: .ASCIZ <15><12>/NO DRIVES AVAILABLE FOR TESTING. PROGRAM ABORTED/

12381	050146	046102	020105	047506
12382	050154	020122	042524	052123
12383	050162	047111	027107	050040
12384	050170	047522	051107	046501
12385	050176	040440	047502	052122
12386	050204	042105	000	
12387	050207	015	052012	042510
12388	050214	043040	046117	047514
12389	050222	044527	043516	042040
12390	050230	044522	042526	020123
12391	050236	044527	046114	041040
12392	050244	020105	042524	052123
12393	050252	042105	005015	000
12394	050257	015	047012	020117
12395	050264	040520	044522	054524
12396	050272	047440	052120	047511
12397	050300	026516	047125	041111
12398	050306	051525	050040	051101
12399	050314	052111	020131	051105
12400	050322	047522	020122	047101
12401	050330	006504	012	
12402	050333	062	024064	024470
12403	050340	051440	041505	047524
12404	050346	020122	047506	046522
12405	050354	052101	042040	052101
12406	050362	020101	043130	051105
12407	050370	052040	051505	020124
12408	050376	054502	040520	051523
12409	050404	042105	005015	000
12410	050411	015	046412	046505
12411	050416	051117	020131	044523
12412	050424	042532	047040	052117
12413	050432	046040	051101	042507
12414	050440	042440	047516	043525
12415	050446	020110	047506	020122
12416	050454	052502	020123	042101
12417	050462	051104	051505	020123
12418	050470	044502	051524	030440
12419	050476	020066	020046	033461
12420	050504	052040	051505	051524
12421	050512	005015		
12422	050514	046101	020114	040504
12423	050522	040524	054040	042506
12424	050530	020122	042524	052123
12425	050536	020123	044527	044124
12426	050544	040440	042104	020122
12427	050552	000076		
12428	050554	031063	000113	
12429	050560	032066	000113	
12430	050564	033071	000113	
12431	050570	041040	050131	051501
12432	050576	042523	006504	000012
12433	050604	005015	020012	020040
12434	050612	020040	020040	025052
12435	050620	020052	040503	052125
12436	050626	047511	020116	025052

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

OPR010: .ASCII <15><12>/NO PARITY OPTION-UNIBUS PARITY ERROR AND/<15><12>

.ASCIZ /24(8) SECTOR FORMAT DATA XFER TEST BYPASSED/<15><12>

OPR011: .ASCII <15><12>/MEMORY SIZE NOT LARGE ENOUGH FOR BUS ADDRESS BITS 16 & 17 TESTS

.ASCIZ /ALL DATA XFER TESTS WITH ADDR >/

OPR012: .ASCIZ /32K/
OPR013: .ASCIZ /64K/
OPR014: .ASCIZ /96K/
OPR015: .ASCIZ / BYPASSED/<15><12>

OPR016: .ASCII <15><12><12>/ *** CAUTION ***/<15><12><12>

12437	050634	006452	005012	
12438	050640	044124	051511	050040
12439	050646	047522	051107	046501
12440	050654	051440	047510	046125
12441	050662	020104	042502	044040
12442	050670	046101	042524	020104
12443	050676	054502	052040	050131
12444	050704	047111	020107	041536
12445	050712	006456	012	
12446	050715	111	020106	040510
12447	050722	052114	042105	052440
12448	050730	044523	043516	052040
12449	050736	042510	044040	046101
12450	050744	020124	042513	026131
12451	050752	052040	042510	051440
12452	050760	040524	042524	047440
12453	050766	020106	044124	020105
12454	050774	051104	053111	006505
12455	051002	012		
12456	051003	117	020122	040503
12457	051010	052122	044522	043504
12458	051016	020105	040503	047116
12459	051024	052117	041040	020105
12460	051032	051120	042105	041511
12461	051040	042524	027104	005015
12462	051046	012		
12463	051047	101	046114	042040
12464	051054	044522	042526	020123
12465	051062	047524	041040	020105
12466	051070	042524	052123	042105
12467	051076	046440	051525	020124
12468	051104	042502	047440	026516
12469	051112	044514	042516	006454
12470	051120	012		
12471	051121	122	040505	054504
12472	051126	020054	047101	020104
12473	051134	051127	052111	020105
12474	051142	047514	045503	051040
12475	051150	051505	052105	006456
12476	051156	012		
12477	051157	101	054516	042040
12478	051164	044522	042526	047040
12479	051172	052117	052040	020117
12480	051200	042502	052040	051505
12481	051206	042524	020104	052515
12482	051214	052123	041040	020105
12483	051222	043117	026506	044514
12484	051230	042516	006456	005012
12485	051236	047516	042524	020072
12486	051244	047062	020104	047101
12487	051252	020104	052523	051502
12488	051260	050505	042525	052116
12489	051266	050040	051501	020123
12490	051274	052522	020116	044524
12491	051302	042515	044440	006523
12492	051310	012		

.ASCII /THIS PROGRAM SHOULD BE HALTED BY TYPING ↑C./<15><12>

.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>

K02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 230
ASCII MESSAGES

SEQ 0230

12493	051311	040	020040	020040
12494	051316	040440	050120	047522
12495	051324	020130	020067	044515
12496	051332	052516	042524	020123
12497	051340	047506	020122	040505
12498	051346	044103	042040	044522
12499	051354	042526	006456	000012
12500	051362	005015	044506	051522
12501	051370	020124	032462	020066
12502	051376	042523	052103	051117
12503	051404	020123	047516	020124
12504	051412	051502	020105	051105
12505	051420	047522	020122	051106
12506	051426	042505	056	
12507	051431	115	054101	046511
12508	051436	046525	042040	052101
12509	051444	020101	051124	047101
12510	051452	043123	051105	052040
12511	051460	051505	020124	054502
12512	051466	040520	051523	042105
12513	051474	005015	000	
12514	051477	040	020040	020040
12515	051504	005015	047117	054514
12516	051512	030440	042040	044522
12517	051520	042526	020056	053117
12518	051526	051105	040514	050120
12519	051534	042105	047440	042520
12520	051542	040522	044524	047117
12521	051550	020123	054502	040520
12522	051556	051523	042105	005015
12523	051564	000		
12524	051565	015	051412	047503
12525	051572	042520	020072	044103
12526	051600	020061	052050	044522
12527	051606	024507	020054	032505
12528	051614	026463	035470	041440
12529	051622	031110	020054	032105
12530	051630	026471	020062	040450
12531	051636	020103	047503	050125
12532	051644	042514	020054	031056
12533	051652	027526	046503	051
12534	051657	015	040412	045104
12535	051664	051525	020124	033522
12536	051672	020062	047506	020122
12537	051700	047503	051516	040524
12538	051706	052116	046040	053105
12539	051714	046105	047440	020116
12540	051722	044103	006462	000012
12541	051730	005015	051120	043517
12542	051736	040522	020115	040510
12543	051744	052114	050040	047105
12544	051752	044504	043516	026440
12545	051760	041440	051101	051124
12546	051766	042111	042507	043040
12547	051774	051117	040515	020124
12548	052002	042502	047111	020107

.ASCIZ / APPROX 7 MINUTES FOR EACH DRIVE./<15><12>

OPRO17: .ASCII <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./

.ASCIZ /MAXIMUM DATA TRANSFER TEST BYPASSED/<15><12>

OPRO18: .ASCIZ / /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>

OPRO19: .ASCII <15><12>@SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)@

.ASCIZ <15><12>/ADJUST R72 FOR CONSTANT LEVEL ON CH2/<15><12>

OPRO20: .ASCIZ <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12>

12549	052010	047503	051122	041505
12550	052016	042524	006504	000012
12551	052024	025052	025052	020052
12552	052032	050040	047522	051107
12553	052040	046501	044040	046101
12554	052046	042524	020104	025040
12555	052054	025052	025052	005015
12556	052062	000		
12557	052063	103	051101	051124
12558	052070	042111	042507	043040
12559	052076	051117	040515	020124
12560	052104	047503	051122	041505
12561	052112	044524	047117	043040
12562	052120	044501	042514	006504
12563	052126	000012		
12564	052130	005015	051120	043517
12565	052136	040522	020115	041101
12566	052144	051117	044524	043516
12567	052152	041040	041505	052501
12568	052160	042523	042440	051122
12569	052166	051117	052040	051110
12570	052174	051505	047510	042114
12571	052202	042440	041530	042505
12572	052210	042504	006504	000012
12573				
12574				
12575	052216	040506	040524	026514
12576	052224	047516	020116	054105
12577	052232	051511	040524	052116
12578	052240	046440	046505	051117
12579	052246	020131	052101	051040
12580	052254	033113	030461	041040
12581	052262	051501	020105	042101
12582	052270	051104	051505	000123
12583	052276	040506	040524	026514
12584	052304	051127	052111	020105
12585	052312	042522	042101	020131
12586	052320	047101	020104	042511
12587	052326	042040	042111	047040
12588	052334	052117	041440	052501
12589	052342	042523	044440	052116
12590	052350	051105	052522	052120
12591	052356	000		
12592	052357	106	052101	046101
12593	052364	050055	051101	052111
12594	052372	027131	051105	047522
12595	052400	020122	051124	050101
12596	052406	020056	041520	040440
12597	052414	020124	051105	047522
12598	052422	020122	020075	000
12599	052427	105	050130	041505
12600	052434	042524	020104	047111
12601	052442	042524	051122	050125
12602	052450	020124	044504	020104
12603	052456	047516	020124	041517
12604	052464	052503	020122	051117

OPRO21: .ASCIZ /***** PROGRAM HALTED *****/<15><12>

OPRO22: .ASCIZ /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>

ABORT: .ASCIZ <15><12>/PROGRAM ABORTING BECAUSE ERROR THRESHOLD EXCEEDED/<15><12>

.SBTTL ERROR MESSAGES

EM1: .ASCIZ /FATAL-NON EXISTANT MEMORY AT RK611 BASE ADDRESS/

EM2: .ASCIZ /FATAL-WRITE READY AND IE DID NOT CAUSE INTERRUPT/

EM3: .ASCIZ /FATAL-PARITY ERROR TRAP. PC AT ERROR = /

EM4: .ASCIZ /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS: /

12605	052472	053440	051501	046040	
12606	052500	052101	027105	041440	
12607	052506	046517	040515	042116	
12608	052514	053440	051501	000072	
12609	052522	052523	051502	051531	EM5: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET ERROR/
12610	052530	042524	020115	046103	
12611	052536	040505	020122	044504	
12612	052544	020104	047516	020124	
12613	052552	042522	042523	020124	
12614	052560	051105	047522	000122	
12615	052566	052523	051502	051531	EM6: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
12616	052574	042524	020115	046103	
12617	052602	040505	020122	044504	
12618	052610	020104	047516	020124	
12619	052616	042522	042523	020124	
12620	052624	042504	044526	042503	
12621	052632	044440	052116	051105	
12622	052640	052522	052120	000	
12623	052645	123	041125	054523	EM7: .ASCIZ /SUBSYSTEM CLEAR DID NOT SET READY/
12624	052652	052123	046505	041440	
12625	052660	042514	051101	042040	
12626	052666	042111	047040	052117	
12627	052674	051440	052105	051040	
12628	052702	040505	054504	000	
12629	052707	127	051117	020104	EM10: .ASCIZ /WORD COUNT INCORRECT/
12630	052714	047503	047125	020124	
12631	052722	047111	047503	051122	
12632	052730	041505	000124		
12633	052734	052502	020123	042101	EM11: .ASCIZ /BUS ADDRESS INCORRECT/
12634	052742	051104	051505	020123	
12635	052750	047111	047503	051122	
12636	052756	041505	000124		
12637	052762	050125	042520	020122	EM11A: .ASCIZ /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
12638	052770	052502	020123	042101	
12639	052776	051104	051505	020123	
12640	053004	044502	051524	044440	
12641	053012	041516	051117	042522	
12642	053020	052103	024040	040502	
12643	053026	033061	020054	033461	
12644	053034	000051			
12645	053036	054503	044514	042116	EM12: .ASCIZ /CYLINDER ADDRESS INCORRECT/
12646	053044	051105	040440	042104	
12647	053052	042522	051523	044440	
12648	053060	041516	051117	042522	
12649	053066	052103	000		
12650	053071	124	040522	045503	EM13: .ASCIZ /TRACK ADDRESS INCORRECT/
12651	053076	040440	042104	042522	
12652	053104	051523	044440	041516	
12653	053112	051117	042522	052103	
12654	053120	000			
12655	053121	123	041505	047524	EM14: .ASCIZ /SECTOR ADDRESS INCORRECT./
12656	053126	020122	042101	051104	
12657	053134	051505	020123	047111	
12658	053142	047503	051122	041505	
12659	053150	027124	000		
12660	053153	116	047117	042455	EM15: .ASCIZ /NON-EXISTANT DRIVE/

12661	053160	044530	052123	047101		
12662	053166	020124	051104	053111		
12663	053174	000105				
12664	053176	047503	052116	047522	EM16:	.ASCIZ /CONTROLLER TIMEOUT/
12665	053204	046114	051105	052040		
12666	053212	046511	047505	052125		
12667	053220	000				
12668	053221	125	044516	020124	EM17:	.ASCIZ /UNIT FIELD ERROR/
12669	053226	044506	046105	020104		
12670	053234	051105	047522	000122		
12671	053242	052515	044514	046120	EM18:	.ASCIZ /MULIPLE DRIVE SELECT/
12672	053250	020105	051104	053111		
12673	053256	020105	042523	042514		
12674	053264	052103	000			
12675	053267	120	047522	051107	EM19:	.ASCIZ /PROGRAMMING ERROR/
12676	053274	046501	044515	043516		
12677	053302	042440	051122	051117		
12678	053310	000				
12679	053311	116	047117	042455	EM20:	.ASCIZ /NON-EXISTANT MEMORY/
12680	053316	044530	052123	047101		
12681	053324	020124	042515	047515		
12682	053332	054522	000			
12683	053335	125	044516	052502	EM21:	.ASCIZ /UNIBUS PARITY ERROR/
12684	053342	020123	040520	044522		
12685	053350	054524	042440	051122		
12686	053356	051117	000			
12687	053361	111	046114	043505	EM22:	.ASCIZ /ILLEGAL FUNCTION/
12688	053366	046101	043040	047125		
12689	053374	052103	047511	000116		
12690	053402	041505	020103	040510	EM23:	.ASCIZ /ECC HARD/
12691	053410	042122	000			
12692	053413	104	052101	020101	EM24:	.ASCIZ /DATA CHECK/
12693	053420	044103	041505	000113		
12694	053426	051127	052111	020105	EM25:	.ASCIZ /WRITE CHECK/
12695	053434	044103	041505	000113		
12696	053442	040504	040524	046040	EM26:	.ASCIZ /DATA LATE/
12697	053450	052101	000105			
12698	053454	050117	051105	052101	EM27:	.ASCIZ /OPERATION INCOMPLETE/
12699	053462	047511	020116	047111		
12700	053470	047503	050115	042514		
12701	053476	042524	000			
12702	053501	110	040505	042504	EM28:	.ASCIZ /HEADER VRC/
12703	053506	020122	051126	000103		
12704	053514	040502	020104	042523	EM29:	.ASCIZ /BAD SECTOR ERROR/
12705	053522	052103	051117	042440		
12706	053530	051122	051117	000		
12707	053535	103	047117	051124	EM30:	.ASCIZ /CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
12708	053542	046117	042514	020122		
12709	053550	042504	042524	052103		
12710	053556	042105	042040	044522		
12711	053564	042526	041040	051525		
12712	053572	050040	051101	052111		
12713	053600	020131	051105	047522		
12714	053606	000122				
12715	053610	042523	045505	044440	EM31:	.ASCIZ /SEEK INCOMPLETE/
12716	053616	041516	046517	046120		

12717	0533624	052105	000105		
12718	0533630	047516	026516	054105	EM32: .ASCIZ /NON-EXECUTABLE DRIVE FUNCTION/
12719	0533636	041505	052125	041101	
12720	0533644	042514	042040	044522	
12721	0533652	042526	043040	047125	
12722	0533660	052103	047511	000116	
12723	0533666	051104	053111	020105	EM33: .ASCIZ /DRIVE DETECTED DRIVE BUS PARITY ERROR/
12724	0533674	042504	042524	052103	
12725	0533702	042105	042040	044522	
12726	0533710	042526	041040	051525	
12727	0533716	050040	051101	052111	
12728	0533724	020131	051105	047522	
12729	0533732	000122			
12730	0533734	047506	046522	052101	EM34: .ASCIZ /FORMAT ERROR/
12731	0533742	042440	051122	051117	
12732	0533750	000			
12733	0533751	104	044522	042526	EM35: .ASCIZ /DRIVE TYPE ERROR/
12734	0533756	052040	050131	020105	
12735	0533764	051105	047522	000122	
12736	0533772	041501	046040	053517	EM36: .ASCIZ /AC LOW/
12737	054000	000			
12738	054001	123	044520	042116	EM37: .ASCIZ /SPINDLE SPEED LOSS/
12739	054006	042514	051440	042520	
12740	054014	042105	046040	051517	
12741	054022	000123			
12742	054024	051104	053111	020105	EM38: .ASCIZ /DRIVE OFF TRACK/
12743	054032	043117	020106	051124	
12744	054040	041501	000113		
12745	054044	054503	044514	042116	EM39: .ASCIZ /CYLINDER OVERFLOW/
12746	054052	051105	047440	042526	
12747	054060	043122	047514	000127	
12748	054066	046111	042514	040507	EM40: .ASCIZ /ILLEGAL DISK PACK ADDRESS/
12749	054074	020114	044504	045523	
12750	054102	050040	041501	020113	
12751	054110	042101	051104	051505	
12752	054116	000123			
12753	054120	051127	052111	020105	EM41: .ASCIZ /WRITE LOCK ERROR/
12754	054126	047514	045503	042440	
12755	054134	051122	051117	000	
12756	054141	104	044522	042526	EM42: .ASCIZ /DRIVE TIMING ERROR/
12757	054146	052040	046511	047111	
12758	054154	020107	051105	047522	
12759	054162	000122			
12760	054164	047516	041440	051105	EM43: .ASCIZ /NO CERR WITH SOME OTHER ERROR SET/
12761	054172	020122	044527	044124	
12762	054200	051440	046517	020105	
12763	054206	052117	042510	020122	
12764	054214	051105	047522	020122	
12765	054222	042523	000124		
12766	054226	051104	053111	020105	EM44: .ASCIZ /DRIVE UNSAFE/
12767	054234	047125	040523	042506	
12768	054242	000			
12769	054243	103	051105	020122	EM45: .ASCIZ /CERR SET BUT NO OTHER ERROR SET/
12770	054250	042523	020124	052502	
12771	054256	020124	047516	047440	
12772	054264	044124	051105	042440	

12773	054272	051122	051117	051440	
12774	054300	052105	000		
12775					
12776	054303	126	020126	044504	EM46: .ASCIZ /VV DID NOT SET WITH PACK ACK/
12777	054310	020104	047516	020124	
12778	054316	042523	020124	044527	
12779	054324	044124	050040	041501	
12780	054332	020113	041501	000113	
12781	054340	052123	052101	051525	EM47: .ASCIZ /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
12782	054346	053040	046101	042111	
12783	054354	051440	052105	047440	
12784	054362	020116	042523	042514	
12785	054370	052103	052040	020117	
12786	054376	047516	026516	054105	
12787	054404	051511	040524	052116	
12788	054412	042040	044522	042526	
12789	054420	000			
12790	054421	123	040524	052524	EM48: .ASCIZ /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
12791	054426	020123	040526	044514	
12792	054434	020104	042522	042523	
12793	054442	020124	047117	051440	
12794	054450	046105	041505	020124	
12795	054456	047524	042440	044530	
12796	054464	052123	047101	020124	
12797	054472	051104	053111	000105	
12798	054500	052123	052101	051525	EM49: .ASCIZ /STATUS VALID NOT RESET ON DRIVE RELEASE/
12799	054506	053040	046101	042111	
12800	054514	047040	052117	051040	
12801	054522	051505	052105	047440	
12802	054530	020116	051104	053111	
12803	054536	020105	042522	042514	
12804	054544	051501	000105		
12805	054550	054105	042520	052103	EM50: .ASCIZ /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
12806	054556	042105	031040	042116	
12807	054564	044440	052116	051105	
12808	054572	052522	052120	042040	
12809	054600	042111	047040	052117	
12810	054606	047440	041503	051125	
12811	054614	047440	020122	040527	
12812	054622	020123	040514	042524	
12813	054630	020056	047503	046515	
12814	054636	047101	020104	040527	
12815	054644	035123	000		
12816	054647	103	047101	047516	EM51: .ASCII /CANNOT READ BAD SECTOR FILE/<15><12>
12817	054654	020124	042522	042101	
12818	054662	041040	042101	051440	
12819	054670	041505	047524	020122	
12820	054676	044506	042514	005015	
12821	054704	042524	052123	047111	.ASCIZ /TESTING ABORTED ON THIS DRIVE/
12822	054712	020107	041101	051117	
12823	054720	042524	020104	047117	
12824	054726	052040	044510	020123	
12825	054734	051104	053111	000105	
12826	054742	046101	043511	046516	EM52: .ASCIZ /ALIGNMENT PACK ON THIS DRIVE. TESTING ABORTED/
12827	054750	047105	020124	040520	
12828	054756	045503	047440	020116	

12829	054764	044124	051511	042040	
12830	054772	044522	042526	020056	
12831	055000	042524	052123	047111	
12832	055006	020107	041101	051117	
12833	055014	042524	000104		
12834	055020	047524	046440	047101	EMS3: .ASCIZ /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
12835	055026	020131	040502	020104	
12836	055034	042523	052103	051117	
12837	055042	027123	052040	051505	
12838	055050	044524	043516	040440	
12839	055056	047502	052122	042105	
12840	055064	047440	020116	044124	
12841	055072	051511	042040	044522	
12842	055100	042526	000		
12843	055103	110	040505	042504	EMS4: .ASCIZ /HEADER WORD MISCOMPARE/
12844	055110	020122	047527	042122	
12845	055116	046440	051511	047503	
12846	055124	050115	051101	000105	
12847	055132	040504	040524	053440	EMS5: .ASCIZ /DATA WORD MISCOMPARE/
12848	055140	051117	020104	044515	
12849	055146	041523	046517	040520	
12850	055154	042522	000		
12851					
12852	055157	103	047101	047516	EMS6: .ASCIZ /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
12853	055164	020124	044506	042116	
12854	055172	044040	040505	042504	
12855	055200	020122	020060	047111	
12856	055206	051040	040505	020104	
12857	055214	047101	020104	047523	
12858	055222	052122	044040	040505	
12859	055230	042504	051522	000	
12860	055235	123	040524	052524	EMSVAL: .ASCIZ /STATUS VALID/
12861	055242	020123	040526	044514	
12862	055250	000104			
12863	055252	042523	042514	052103	EMNZPR: .ASCIZ /SELECT WITH NON-ZERO STATUS PAIR/
12864	055260	053440	052111	020110	
12865	055266	047516	026516	042532	
12866	055274	047522	051440	040524	
12867	055302	052524	020123	040520	
12868	055310	051111	000		
12869	055313	127	044522	044524	EMWCS2: .ASCIZ /WRITING COMMAND STATUS REGISTER 2/
12870	055320	043516	041440	046517	
12871	055326	040515	042116	051440	
12872	055334	040524	052524	020123	
12873	055342	042522	044507	052123	
12874	055350	051105	031040	000	
12875		053751			EMDTPE= EM35 ;DRIVE TYPE ERROR
12876	055355	104	044522	042526	EMDI: .ASCIZ /DRIVE INTERRUPT/
12877	055362	044440	052116	051105	
12878	055370	052522	052120	000	
12879		053666			EMDPAR= EM33 ;DRIVE DETECTED DRIVE BUS PARITY ERROR
12880	055375	104	044522	042526	EMDSC: .ASCIZ /DRIVE STATUS CHANGE/
12881	055402	051440	040524	052524	
12882	055410	020123	044103	047101	
12883	055416	042507	000		
12884	055421	104	044522	042526	EMDA: .ASCIZ /DRIVE ATTENTION/

E03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 237
ERROR MESSAGES

SEQ 0237

12885	055426	040440	052124	047105	
12886	055434	044524	047117	000	
12887	055441	103	047117	051124	EMCCLR: .ASCIZ /CONTROLLER CLEAR/
12888	055446	046117	042514	020122	
12889	055454	046103	040505	000122	
12890		046576			EMSELD= OPER00 ;DRIVE SELECT
12891		046724			EMDCLR= OPER04 ;DRIVE CLEAR
12892		046765			EMRCAL= OPER12 ;RECALIBRATE
12893	055462	042523	047503	042116	EM2INT: .ASCIZ /SECOND INTERRUPT/
12894	055470	044440	052116	051105	
12895	055476	052522	052120	000	
12896	055503	123	042505	020113	EMSKSF: .ASCIZ /SEEK TO SELF/
12897	055510	047524	051440	046105	
12898	055516	000106			
12899		047010			EMSK= OPER16 ;SEEK
12900	055520	047125	054105	042520	EMUXIT: .ASCIZ /UNEXPECTED INTERRUPT/
12901	5526	052103	042105	044440	
12902	5534	052116	051105	052522	
12903	055542	052120	000		
12904	055545	125	044516	052502	EMUR: .ASCIZ /UNIBUS RESET/
12905	055552	020123	042522	042523	
12906	055560	000124			
12907	055562	042522	042523	000124	EMRSET: .ASCIZ /RESET/
12908		053442			EMDLT= EM26 ;DATA LATE
12909	055570	042522	042101	047111	EMRDB: .ASCIZ /READING DATA BUFFER/
12910	055576	020107	040504	040524	
12911	055604	041040	043125	042506	
12912	055612	000122			
12913	055614	047503	052116	047522	EMCERR: .ASCIZ /CONTROLLER ERROR/
12914	055622	046114	051105	042440	
12915	055630	051122	051117	000	
12916	055635	123	041125	054523	EMSCLR: .ASCIZ /SUBSYSTEM CLEAR/
12917	055642	052123	046505	041440	
12918	055650	042514	051101	000	
12919	055655	115	046125	044524	EMMI: .ASCIZ /MULTIPLE INTERRUPTS/
12920	055662	046120	020105	047111	
12921	055670	042524	051122	050125	
12922	055676	051524	000		
12923	055701	102	042101	051440	DRVABT: .ASCII /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))<15><12>
12924	055706	041505	047524	051522	
12925	055714	047440	020116	040520	
12926	055722	045503	044440	020116	
12927	055730	051101	040505	020123	
12928	055736	051525	042105	041040	
12929	055744	020131	042524	052123	
12930	055752	024040	054503	020114	
12931	055760	030463	024062	024470	
12932	055766	006451	012		
12933	055771	124	051505	044524	.ASCIZ /TESTING ABORTED ON THIS DRIVE/
12934	055776	043516	040440	047502	
12935	056004	052122	042105	047440	
12936	056012	020116	044124	051511	
12937	056020	042040	044522	042526	
12938	056026	000			
12939					
12940					.SBTTL DATA HEADERS FOR ERROR REPORTS

F03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 238
DATA HEADERS FOR ERROR REPORTS

SEQ 0238

12941	056027	124	052123	047040	DH001: .ASCIZ /TST NUM ERR PC DRIVE/
12942	056034	046525	042440	051122	
12943	056042	050040	020103	042040	
12944	056050	044522	042526	000	
12945	056055	124	052123	047040	DH002A: .ASCIZ /TST NUM ERR PC DRIVE/
12946	056062	046525	042440	051122	
12947	056070	050040	020103	042040	
12948	056076	044522	042526	000	
12949	056103	122	041513	030523	DH002B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKDCYL RKDA/
12950	056110	020040	051040	041513	
12951	056116	031123	020040	051040	
12952	056124	042113	020123	020040	
12953	056132	051040	042513	020122	
12954	056140	020040	051040	040513	
12955	056146	047523	020106	051040	
12956	056154	042113	054503	020114	
12957	056162	051040	042113	000101	
12958	056170	045522	040502	020040	DH002C: .ASCIZ /RKBA RKWC/
12959	056176	020040	045522	041527	
12960	056204	000			
12961	056205	101	030060	020040	DH002D: .ASCIZ /A00 B00 A01 B01 A02 B02 A03 B03/
12962	056212	020040	041040	030060	
12963	056220	020040	020040	040440	
12964	056226	030460	020040	020040	
12965	056234	041040	030460	020040	
12966	056242	020040	040440	031060	
12967	056250	020040	020040	041040	
12968	056256	031060	020040	020040	
12969	056264	040440	031460	020040	
12970	056272	020040	041040	031460	
12971	056300	000			
12972	056301	122	041513	030523	DH003B: .ASCIZ /RKCS1 RKCS2 RKDS RKER RKASOF RKMR1/
12973	056306	020040	051040	041513	
12974	056314	031123	020040	051040	
12975	056322	042113	020123	020040	
12976	056330	051040	042513	020122	
12977	056336	020040	051040	040513	
12978	056344	047523	020106	051040	
12979	056352	046513	030522	000	
12980	056357	124	042510	040440	DH005: .ASCIZ /THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:/
12981	056364	047502	042526	040440	
12982	056372	042522	042440	050130	
12983	056400	041505	042524	020104	
12984	056406	051105	047522	051522	
12985	056414	052040	040510	020124	
12986	056422	044504	020104	047516	
12987	056430	020124	042523	020124	
12988	056436	047111	047440	042520	
12989	056444	040522	044524	047117	
12990	056452	000072			
12991	056454	044124	020105	041101	DH006: .ASCIZ /THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:/
12992	056462	053117	020105	051101	
12993	056470	020105	047125	054105	
12994	056476	042520	052103	042105	
12995	056504	042440	051122	051117	
12996	056512	020123	042523	020124	

12997	056520	047111	047440	042520	
12998	056526	040522	044524	047117	
12999	056534	000072			
13000	056536	044124	020105	041101	DH007: .ASCIZ /THE ABOVE ARE ERRORS SET IN OPERATION:/
13001	056544	053117	020105	051101	
13002	056552	020105	051105	047522	
13003	056560	051522	051440	052105	
13004	056566	044440	020116	050117	
13005	056574	051105	052101	047511	
13006	056602	035116	000		
13007	056605	101	054516	043040	DH005A: .ASCIZ /ANY FIELD WITH ALL ONES MUST BE CONSIDERED INVALID/
13008	056612	042511	042114	053440	
13009	056620	052111	020110	046101	
13010	056626	020114	047117	051505	
13011	056634	046440	051525	020124	
13012	056642	042502	041440	047117	
13013	056650	044523	042504	042522	
13014	056656	020104	047111	040526	
13015	056664	044514	000104		
13016	056670	051105	047522	020122	DH010: .ASCIZ /ERROR AT COMPLETION OF OPERATION/
13017	056676	052101	041440	046517	
13018	056704	046120	052105	047511	
13019	056712	020116	043117	047440	
13020	056720	042520	040522	044524	
13021	056726	047117	000		
13022	056731	105	050130	042124	DH010A: .ASCIZ /EXPTD IS/
13023	056736	020040	044440	000123	
13024	056744	042522	042101	042040	DH010B: .ASCIZ /READ DATA WITH FORCED CYLINDER OVERFLOW/
13025	056752	052101	020101	044527	
13026	056760	044124	043040	051117	
13027	056766	042503	020104	054503	
13028	056774	044514	042116	051105	
13029	057002	047440	042526	043122	
13030	057010	047514	000127		
13031	057014	047516	020124	042523	DH011: .ASCIZ /NOT SET AS A RESULT OF/
13032	057022	020124	051501	040440	
13033	057030	051040	051505	046125	
13034	057036	020124	043117	000	
13035	057043	116	052117	051040	DH012: .ASCIZ /NOT RESET AS A RESULT OF/
13036	057050	051505	052105	040440	
13037	057056	020123	020101	042522	
13038	057064	052523	052114	047440	
13039	057072	000106			
13040	057074	042523	020124	051501	DH013: .ASCIZ /SET AS A RESULT OF/
13041	057102	040440	051040	051505	
13042	057110	046125	020124	043117	
13043	057116	000			
13044	057117	122	051505	052105	DH014: .ASCIZ /RESET AS A RESULT OF/
13045	057124	040440	020123	020101	
13046	057132	042522	052523	052114	
13047	057140	047440	000106		
13048	057144	047507	042117	020040	DH015: .ASCIZ /GOOD BAD WORD NUM/
13049	057152	020040	040502	020104	
13050	057160	020040	020040	047527	
13051	057166	042122	047040	046525	
13052	057174	000			

H03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 240
DATA HEADERS FOR ERROR REPORTS

SEQ 0240

13053	057175	123	041125	054523	DH016: .ASCIZ /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/
13054	057202	052123	046505	041440	
13055	057210	042514	051101	052040	
13056	057216	020117	042522	042523	
13057	057224	020124	042523	045505	
13058	057232	046040	046511	052111	
13059	057240	040440	046114	053517	
13060	057246	047111	020107	042510	
13061	057254	042101	020123	047524	
13062	057262	051040	046105	040517	
13063	057270	000104			
13064	057272	042523	042514	052103	DH017: .ASCIZ /SELECT AFTER RECAL/
13065	057300	040440	052106	051105	
13066	057306	051040	041505	046101	
13067	057314	000			
13068	057315	122	041505	046101	DH018: .ASCIZ /RECAL COMPLETE ATTN AFTER SELECT/
13069	057322	041440	046517	046120	
13070	057330	052105	020105	052101	
13071	057336	047124	040440	052106	
13072	057344	051105	051440	046105	
13073	057352	041505	000124		
13074					
13075					.SBTTL DATA TABLES FOR ERROR REPORTS
13076					.EVEN
13077	057356	001302	001116	001626	DT003: .WORD \$TESTN,\$ERRPC,DRVNUM,T.CS1,T.CS2,T.DS,T.ER,T.ASOF,T.MR1
13078	057364	001540	001550	001552	
13079	057372	001554	001556	001566	
13080	057400	001302	001116	001626	DT001: .WORD \$TESTN,\$ERRPC,DRVNUM
13081	057406				DT002:
13082	057406				DT004:
13083	057406				DT005:
13084	057406				DT006:
13085	057406				DT007:
13086	057406	001302	001116	001626	DT010: .WORD \$TESTN,\$ERRPC,DRVNUM
13087	057414	001540	001550	001552	.WORD T.CS1,T.CS2,T.DS,T.ER,T.ASOF
13088	057422	001554	001556		
13089	057426	001560	001546		.WORD T.DCYL,T.DA
13090	057432	001544	001542		.WORD T.BA,T.WC
13091	057436	001202	001204	001206	.WORD \$REG10,\$REG11,\$REG12,\$REG13,\$REG14,\$REG15,\$REG16,\$REG17
13092	057444	001210	001212	001214	
13093	057452	001216	001220		
13094	057456	001302	001116	001626	DT015: .WORD \$TESTN,\$ERRPC,DRVNUM
13095	057464	001202	001204	001206	DT015A: .WORD \$REG10,\$REG11,\$REG12
13096					.SBTTL DATA FORMATS FOR ERROR REPORTS
13097	057472	000001			DF001: .WORD 1
13098	057474	003	000		.BYTE 3,0
13099					
13100	057476	000004			DF002: .WORD 4
13101	057500	000	000		.BYTE 0,0
13102	057502	056055			.WORD DH002A
13103	057504	003	000		.BYTE 3,0
13104	057506	056103			.WORD DH002B
13105	057510	007	000		.BYTE 7,0
13106	057512	056170			.WORD DH002C
13107	057514	002	000		.BYTE 2,0
13108					

13109					
13110	057516	000002		DF003:	.WORD 2
13111	057520	003	000		.BYTE 3,0
13112	057522	056301			.WORD DH003B
13113	057524	006	000		.BYTE 6,0
13114					
13115	057526	000006		DF004:	.WORD 6
13116	057530	000	000		.BYTE 0,0
13117	057532	000000		DF004A:	.WORD 0
13118	057534	000	000		.BYTE 0,0
13119	057536	056055			.WORD DH002A
13120	057540	003	000		.BYTE 3,0
13121	057542	056103			.WORD DH002B
13122	057544	007	000		.BYTE 7,0
13123	057546	056170			.WORD DH002C
13124	057550	002	000		.BYTE 2,0
13125	057552	056205			.WORD DH002D
13126	057554	010	000		.BYTE 10,0
13127					
13128	057556	000007		DF005:	.WORD 7
13129	057560	000	000		.BYTE 0,0
13130	057562	000000		DF005A:	.WORD 0
13131	057564	000	000		.BYTE 0,0
13132	057566	056055			.WORD DH002A
13133	057570	003	000		.BYTE 3,0
13134	057572	056103			.WORD DH002B
13135	057574	007	000		.BYTE 7,0
13136	057576	056170			.WORD DH002C
13137	057600	002	000		.BYTE 2,0
13138	057602	056205			.WORD DH002D
13139	057604	010	000		.BYTE 10,0
13140	057606	056605			.WORD DH005A
13141	057610	000	000		.BYTE 0,0
13142					
13143	057612	000005		DF006:	.WORD 5
13144	057614	000	000		.BYTE 0,0
13145	057616	000000		DF006A:	.WORD 0
13146	057620	000	000		.BYTE 0,0
13147	057622	056055			.WORD DH002A
13148	057624	003	000		.BYTE 3,0
13149	057626	056103			.WORD DH002B
13150	057630	007	000		.BYTE 7,0
13151	057632	056170			.WORD DH002C
13152	057634	002	000		.BYTE 2,0
13153					
13154	057636	000004		DF007:	.WORD 4
13155	057640	000	000		.BYTE 0,0
13156	057642	000000		DF007A:	.WORD 0
13157	057644	000	000		.BYTE 0,0
13158	057646	056055			.WORD DH002A
13159	057650	003	000		.BYTE 3,0
13160	057652	056301			.WORD DH003B
13161	057654	006	000		.BYTE 6,0
13162					
13163	057656	000004		DF010:	.WORD 4
13164	057660	000	000		.BYTE 0,0

13165	057662	000000		DF010A: .WORD	0
13166	057664	000	000	.BYTE	0,0
13167	057666	056055		.WORD	DH002A
13168	057670	003	000	.BYTE	3,0
13169	057672	056731		.WORD	DH010A
13170	057674	002	000	.BYTE	2,0
13171					
13172	057676	000004		DF011: .WORD	4
13173	057700	000	000	.BYTE	0,0
13174	057702	000000		DF011A: .WORD	0
13175	057704	000	000	.BYTE	0,0
13176	057706	056055		.WORD	DH002A
13177	057710	003	000	.BYTE	3,0
13178	057712	056301		.WORD	DH003B
13179	057714	000006	000000	.WORD	6,0
13180					
13181	057720	000002		DF015: .WORD	2
13182	057722	003	000	.BYTE	3,0
13183	057724	057144		.WORD	DH015
13184	057726	003	000	.BYTE	3,0
13185					
13186	057730	000001		DF016: .WORD	1
13187	057732	003	000	.BYTE	3,0
13188					
13189	057734	000052		BS26: .BLKW	52
13190	060060	000052		BS24: .BLKW	52
13191	060204	000200		PATCH: .BLKW	200
13192	060604	001000		IBUFF: .BLKW	1000
13193	062604	001000		OBUFF: .BLKW	1000
13194		000001		.END	

ABASE = 177440	1956#	2278	2319
ABORT 052130	9449	12564#	
ABTFAI 042705	9244	11170#	
ACDW1 = 000000	2278	2321	
ACDW2 = 000000	2278		
ACLO = 000010	2051#	10266	
ACLOER= 000100	10268	12135#	
ACPUOP= 000000	2278	2293	
ADDWD = 000000	2278		
ADDW1 = 000000	2278		
ADDW10= 000000	2278		
ADDW11= 000000	2278		
ADDW12= 000000	2278		
ADDW13= 000000	2278		
ADDW14= 000000	2278		
ADDW15= 000000	2278		
ADDW2 = 000000	2278		
ADDW3 = 000000	2278		
ADDW4 = 000000	2278		
ADDW5 = 000000	2278		
ADDW6 = 000000	2278		
ADDW7 = 000000	2278		
ADDW8 = 000000	2278		
ADDW9 = 000000	2278		
ADEVCT= 000000	2278	2284	
ADEVM = 000000	2278	2320	
ADJCLK 042332	3185	11063#	
AENV = 000000	2278	2289	
AENVM = 000000	2278	2290	
AFATAL= 000000	2278	2281	
AMADR1= 000000	2278	2306	
AMADR2= 000000	2278	2310	
AMADR3= 000000	2278	2313	
AMADR4= 000000	2278	2316	
AMAMS1= 000000	2278	2300	
AMAMS2= 000000	2278	2308	
AMAMS3= 000000	2278	2311	
AMAMS4= 000000	2278	2314	
AMSGAD= 000000	2278	2286	
AMSGLG= 000000	2278	2287	
AMSGTY= 000000	2278	2280	
AMTYP1= 000000	2278	2301	
AMTYP2= 000000	2278	2309	
AMTYP3= 000000	2278	2312	
AMTYP4= 000000	2278	2315	
APASS = 000000	2278	2283	
APRIOR= 000240	1955#	2278	
APTCSU= 000040	9302#	11203	
APTENV= 000001	9258	9300#	9357 11196
APTSIZ= 000200	2775	9299#	
APTSP0= 000100	9260	9301#	11198
ASWREG= 000000	2278	2291	
ATESTN= 000000	2278	2282	
AUNIT = 000000	2278	2285	
AUSWR = 000000	2278	2292	
AVECT1= 000210	1954#	2278	2317

SW00	=	000001	1866#	1876
SW01	=	000002	1865#	1875
SW02	=	000004	1864#	1874
SW03	=	000010	1863#	1873
SW04	=	000020	1862#	1872
SW05	=	000040	1861#	1871
SW06	=	000100	1860#	1870
SW07	=	000200	1859#	1869
SW08	=	000400	1858#	1868
SW09	=	001000	1857#	1867
SW1	=	000002	1875#	
SW10	=	002000	1856#	
SW11	=	004000	1855#	
SW12	=	010000	1854#	9445
SW13	=	020000	1853#	10507
SW14	=	040000	1852#	
SW15	=	100000	1851#	
SW2	=	000004	1874#	
SW3	=	000010	1873#	
SW4	=	000020	1872#	
SW5	=	000040	1871#	
SW6	=	000100	1870#	
SW7	=	000200	1869#	
SW8	=	000400	1868#	
SW9	=	001000	1867#	9551
S.ACLO	=	000100	2095#	
S.BRHM	=	000100	2109#	
S.BRKE	=	040000	2131#	
S.CART	=	000400	2111#	
S.DIB	=	002000	2127#	
S.DOOR	=	000200	2110#	
S.DRA	=	000040	2081#	
S.DROT	=	020000	2102#	
S.DRY	=	000200	2083#	
S.DSC	=	040000	2090#	
S.FLT	=	000200	2096#	
S.FORM	=	001000	2085#	
S.FWD	=	002000	2113#	
S.HDFL	=	000200	2124#	
S.HDHM	=	000040	2108#	8370
S.ICYL	=	000040	2094#	
S.ILF	=	000400	2097#	
S.LIMD	=	020000	2130#	
S.LOAD	=	010000	2115#	
S.MHD	=	000400	2125#	
S.NMOV	=	010000	2129#	
S.OFF	=	002000	2086#	
S.PAR	=	001000	2098#	
S.PIP	=	020000	2089#	
S.PLO	=	004000	2128#	
S.REV	=	004000	2114#	
S.RTZ	=	020000	2116#	
S.SECT	=	000020	2121#	
S.SKI	=	002000	2099#	
S.SPIN	=	010000	2088#	
S.SPLS	=	010000	2101#	

L04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 258
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0257

TST1	003116	2896#	9010	9170
TST10	005060	3373#	9177	
TST100	025306	7847#	9233	
TST101	025500	7927#	9234	
TST102	025724	8019#	9235	
TST103	026052	8075#	9236	
TST104	026376	8228#	9237	
TST105	027130	8435#	9238	
TST106	027332	8493#	9239	
TST107	027552	8562#	9240	
TST11	006004	3551#	9178	
TST110	027766	8635#	9241	
TST111	030464	8799#	9242	
TST12	006122	3599#	9179	
TST13	006232	3649#	9180	
TST14	006474	3715#	9181	
TST15	006626	3757#	9182	
TST16	007036	3818#	9183	
TST17	007226	3881#	9184	
TST2	003234	2927#	9171	
TST20	007350	3944#	9185	
TST21	010136	4086#	9186	
TST22	010534	4213#	9187	
TST23	010746	4289#	9188	
TST24	011164	4368#	9189	
TST25	011310	4420#	9190	
TST26	011442	4474#	9191	
TST27	011620	4534#	9192	
TST3	003322	2987#	9172	
TST30	011752	4593#	9193	
TST31	012246	4690#	9194	
TST32	012442	4765#	9195	
TST33	012676	4846#	9196	
TST34	013064	4914#	9197	
TST35	013316	4995#	9198	
TST36	013504	5065#	9199	
TST37	013726	5145#	9200	
TST4	004346	3196#	9173	
TST40	014114	5214#	9201	
TST41	014306	5289#	9202	
TST42	014502	5358#	9203	
TST43	014706	5411#	9204	
TST44	015112	5472#	9205	
TST45	015572	5582#	9206	
TST46	015776	5666#	9207	
TST47	016202	5752#	9208	
TST5	004450	3236#	9174	
TST50	016420	5837#	9209	
TST51	016624	5922#	9210	
TST52	017030	6009#	9211	
TST53	017154	6068#	9212	
TST54	017300	6125#	9213	
TST55	017424	6182#	9214	
TST56	017550	6238#	9215	
TST57	017674	6291#	9216	
TST6	004626	3286#	9175	

E05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 264
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0263

		3716*	3758*	3819*	3882*	3945*	4087*	4214*	4290*	4369*	4421*	4475*	4535*	4594*
		4691*	4766*	4847*	4915*	4996*	5066*	5146*	5215*	5290*	5359*	5412*	5473*	5583*
		5567*	5753*	5838*	5923*	6010*	6069*	6126*	6183*	6239*	6292*	6319*	6354*	6433*
		6507*	6575*	6654*	6808*	6883*	6939*	7072*	7199*	7261*	7463*	7630*	7721*	7810*
		7848*	7928*	8020*	8076*	8229*	8436*	8469*	8494*	8563*	8636*	8800*	8976*	9148*
		9155	9158*	9168										
STKB	001146	2227*	11399	11420	11431	11456	11484	11511						
STKCN	043654	11400*	11415*	11445	11462*	11576	11578*							
STKINT	043664	2734	3780	8537	11415*	11436	11497							
STKGEN=	043663	11404*	11470	11581										
STKQIN	043656	11401*	11416*	11417	11468*	11469*	11470	11472*						
STKQOU	043660	11402*	11417*	11579	11580*	11581	11583*							
STKQSR	043662	11403*	11416	11472	11583									
STKS	001144	2226*	11399	11421*	11452*	11454	11460*	11482	11498*	11508	11520*	11540*		
STKSRV	043734	11418	11431*											
STMP0	001222	2252*	2834*	2836	2838	2845*	2847	2850	2862*	2864	2866	2868*	2869*	2870*
		2871*	2872*	2874	9979*	9983*								
STMP1	001224	2253*	8647*	8723*	8748	8759	8781*	8802*	8903*	8928	8939	8961*	10405	10784*
		10786*	10788	10802*	10818	10819	10823	10825*						
STMP10	001242	2260*	10179*	10359	10461	10591								
STMP11	001244	2261*	10180*	10440	10585									
STMP12	001246	2262*	10181*	10420	10580									
STMP13	001250	2263*	10183*	10187*	10217	10356	10374	10418	10438	10459	10510	10565		
STMP14	001252	2264*	10533*	10552	10556*	10558	10562*	10583	10587*	10589	10593*			
STMP15	001254	2265*	10534*	10539*										
STMP16	001256	2266*	9763*	9768*										
STMP17	001260	2267*	9765*	9773	9775*									
STMP2	001226	2254*	10406											
STMP3	001230	2255*	9918*	9932*	9947									
STMP4	001232	2256*	3015*	3034	3038*									
STMP5	001234	2257*	10641*	10651*	10658									
STMP6	001236	2258*	10638*	10659										
STMP7	001240	2259*	3114*	3178*	3181	10639*	10660							
STN =	000112	1786*	1797	2889	2897*	2920	2928*	2950	2988*	3189	3197*	3226	3237*	3278
		3287*	3322	3330*	3355	3374*	3541	3552*	3588	3600*	3636	3650*	3707	3716*
		3750	3758*	3808	3819*	3874	3882*	3920	3945*	4077	4087*	4197	4214*	4279
		4290*	4359	4369*	4412	4421*	4464	4475*	4525	4535*	4584	4594*	4681	4691*
		4755	4766*	4836	4847*	4904	4915*	4985	4996*	5055	5066*	5135	5146*	5206
		5215*	5279	5290*	5351	5359*	5404	5412*	5458	5473*	5574	5583*	5657	5667*
		5744	5753*	5828	5838*	5913	5923*	5998	6010*	6057	6069*	6114	6126*	6171
		6183*	6227	6239*	6283	6292*	6310	6319*	6344	6354*	6424	6433*	6493	6507*
		6565	6575*	6639	6654*	6799	6808*	6874	6883*	6920	6939*	7053	7072*	7188
		7199*	7241	7261*	7446	7463*	7620	7630*	7711	7721*	7801	7810*	7836	7848*
		7917	7928*	8008	8020*	8064	8076*	8214	8229*	8423	8436*	8486	8494*	8554
		8563*	8620	8636*	8784	8800*	9130	9170	11091					
STPB	001152	2229*	11241*	11252										
STPFLG	001157	2233*	11190	11252										
STPS	001150	2228*	11239	11252										
STRAP	045524	2748	11801*											
STRAP2	045546	11812*	11823											
STRP =	000041	11816*	11825*	11826*	11827*	11828*	11829*	11830	11831*	11832	11833*	11834*	11835*	11836*
		11837*	11838*	11839*	11840*	11841*	11842*	11843*	11844*	11845*	11846*	11847*	11848*	11849*
		11850*	11851*	11852*	11853*	11854*	11855*	11856*	11857*	11858*				
STRPAD	045560	11806	11823*	11858										
STSTM	000230	2193*												
STSTNM	001102	2206*	8481*	8975*	9095	9132*	9159*	9160	9165	9169	9321	9380	9391	11096

COMMEN	1919#														
ENDCOM	1919#														
ERROR	1813#	2914	2943	2990	3020	3029	3052	3059	3081	3118	3126	3134	3139	3146	3151
	3157	3160	3165	3172	3199	3206	3209	3215	3218	3222	3239	3246	3248	3255	3261
	3264	3271	3289	3296	3299	3306	3317	3332	3340	3346	3353	3376	3384	3390	3404
	3409	3415	3422	3431	3437	3443	3449	3455	3460	3466	3472	3478	3491	3494	3500
	3506	3518	3521	3532	3538	3554	3561	3564	3578	3584	3587	3602	3609	3612	3626
	3632	3652	3659	3666	3674	3681	3688	3695	3699	3705	3718	3725	3734	3746	3749
	3760	3767	3775	3795	3802	3823	3831	3836	3841	3849	3860	3868	3884	3892	3897
	3902	3910	3916	3956	3968	3970	3981	3991	4001	4008	4017	4030	4072	4092	4108
	4111	4114	4133	4136	4139	4147	4154	4162	4165	4166	4167	4173	4182	4217	4240
	4243	4246	4257	4266	4276	4293	4320	4322	4326	4338	4347	4356	4372	4389	4392
	4404	4410	4424	4442	4445	4457	4463	4478	4503	4506	4518	4524	4539	4562	4565
	4576	4579	4597	4615	4618	4645	4648	4660	4663	4669	4677	4694	4711	4714	4734
	4737	4743	4750	4781	4797	4800	4816	4818	4824	4831	4857	4871	4874	4885	4887
	4893	4900	4931	4947	4950	4964	4967	4973	4981	5005	5019	5022	5033	5036	5042
	5050	5082	5097	5100	5112	5115	5122	5130	5155	5170	5173	5184	5187	5193	5201
	5217	5233	5236	5253	5256	5267	5275	5292	5307	5310	5325	5327	5338	5346	5361
	5373	5378	5387	5393	5400	5414	5426	5431	5440	5447	5454	5475	5526	5540	5548
	5555	5562	5568	5585	5601	5604	5620	5623	5635	5638	5644	5652	5669	5685	5688
	5708	5711	5723	5726	5732	5740	5755	5771	5774	5795	5798	5809	5815	5823	5840
	5856	5859	5876	5879	5891	5894	5900	5908	5925	5941	5945	5961	5964	5976	5979
	5985	5993	6012	6035	6038	6044	6052	6071	6092	6095	6101	6109	6128	6150	6153
	6159	6167	6185	6206	6209	6215	6223	6241	6262	6265	6271	6279	6294	6306	6309
	6321	6333	6339	6360	6373	6376	6388	6396	6397	6398	6405	6413	6436	6456	6459
	6471	6474	6480	6488	6537	6548	6551	6579	6599	6602	6614	6617	6623	6631	6657
	6675	6678	6698	6704	6713	6722	6728	6739	6742	6748	6756	6774	6780	6786	6794
	6814	6828	6831	6844	6858	6861	6885	6901	6904	6916	6919	6943	6958	6961	6973
	6976	6982	6993	6999	7018	7026	7035	7043	7076	7091	7094	7106	7109	7116	7128
	7134	7153	7161	7170	7178	7201	7217	7220	7234	7237	7281	7284	7301	7312	7315
	7326	7334	7339	7355	7358	7360	7367	7380	7384	7395	7406	7409	7417	7425	7465
	7484	7493	7498	7510	7514	7517	7520	7532	7535	7556	7560	7570	7574	7577	7579
	7591	7594	7613	7617	7632	7644	7647	7651	7653	7665	7668	7680	7692	7698	7700
	7704	7724	7736	7740	7743	7755	7758	7770	7776	7786	7797	7812	7827	7833	7850
	7866	7869	7902	7908	7913	7931	7943	7946	7958	7976	7981	7985	7990	8002	8004
	8022	8034	8037	8040	8060	8063	8078	8090	8094	8106	8109	8122	8125	8130	8133
	8145	8148	8166	8190	8194	8207	8213	8232	8244	8247	8251	8254	8266	8269	8281
	8284	8296	8324	8327	8341	8347	8350	8365	8368	8374	8381	8386	8400	8403	8418
	8421	8439	8447	8462	8465	8499	8513	8529	8533	8551	8568	8581	8597	8601	8617
	8658	8666	8668	8678	8681	8687	8696	8704	8812	8820	8824	8829	8832	8841	8844
	8848	8851	8862	8874	8877	8884	11094	11110	11113	11117	11120	11123	11145	11148	
ESCAPE	1919#														
GETPRI	1919#	9037													
GETSWR	1919#	2786#													
INITSS	2712#	4106	4131	4160	4693	4779	4855	4929	5003	5080	5153	5291	5360	5413	5474
	5584	5668	5754	5839	5924	6011	6070	6127	6184	6240	6293	6320	6358	6386	6435
	6536	6578	6656	6726	6812	6842	6884	6942	6981	7075	7115	7200	7299	7365	7393
	7464	7631	7699	7723	7757	7785	7811	7849	7930	7983	8021	8039	8077	8231	8349
LOADLS	8373	8380	8438	8446	8498	8567	8656	8810	11093	11122					
	2714#	4696	4716	4807	4858	4955	5006	5102	5157	5294	5312	5485	5587	5610	5625
	5671	5694	5713	5757	5785	5799	5842	5866	5881	5927	5951	5966	6025	6082	6140
	6196	6252	6296	6323	6446	6461	6538	6589	6604	6659	6688	6729	6764	6845	6887
	6906	6948	6963	6983	7081	7096	7118	7207	7224	7302	7345	7368	7469	7500	7522
	7537	7561	7581	7596	7634	7655	7670	7682	7726	7745	7760	7814	7956	7875	7892
	7933	7948	7960	7992	8024	8042	8080	8096	8111	8135	8150	8196	8234	8256	8271
	8286	8298	8329	8352	8390	8408	8452	11100	11129						

H05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR&KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 268
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0266

MSG	2889#	2891	2920#	2922	2949#	2952	3189#	3191	3226#	3228	3278#	3280	3322#	3324	3355#
	3357	3541#	3543	3588#	3590	3636#	3638	3707#	3709	3750#	3752	3808#	3810	3874#	3876
	3920#	3922	4077#	4079	4197#	4199	4279#	4281	4359#	4361	4412#	4414	4464#	4466	4525#
	4527	4584#	4586	4681#	4683	4755#	4757	4836#	4838	4904#	4906	4985#	4987	5055#	5057
	5135#	5137	5206#	5208	5279#	5281	5351#	5353	5404#	5406	5458#	5460	5574#	5576	5657#
	5659	5744#	5746	5828#	5830	5913#	5915	5998#	6000	6057#	6059	6114#	6116	6171#	6173
	6227#	6229	6283#	6285	6310#	6312	6344#	6346	6424#	6426	6493#	6495	6565#	6567	6639#
	6641	6799#	6801	6874#	6876	6920#	6922	7053#	7055	7188#	7190	7241#	7243	7446#	7448
	7620#	7622	7710#	7713	7801#	7803	7835#	7838	7916#	7919	8008#	8010	8064#	8066	8214#
	8216	8423#	8425	8486#	8488	8554#	8556	8620#	8622	8784#	8786				
MULT	1919#														
NEWTST	1919#	2889	2920	2950	3189	3226	3278	3322	3355	3541	3588	3636	3707	3750	3808
	3874	3920	4077	4197	4279	4359	4412	4464	4525	4584	4681	4755	4836	4904	4985
	5055	5135	5206	5279	5351	5404	5458	5574	5657	5744	5828	5913	5998	6057	6114
	6171	6227	6283	6310	6344	6424	6493	6565	6639	6799	6874	6920	7053	7188	7241
	7446	7620	7711	7801	7836	7917	8008	8064	8214	8423	8486	8554	8620	8784	
OPCHK	2716#	4713	4736	4799	4817	4873	4886	4949	4966	5021	5035	5099	5114	5172	5186
	5309	5326	5603	5622	5637	5687	5710	5725	5773	5797	5858	5878	5893	5944	5963
	5978	6037	6094	6152	6208	6264	6308	6375	6458	6473	6550	6601	6616	6677	6741
	6830	6860	6903	6918	6960	6975	7093	7108	7219	7236	7283	7314	7337	7357	7382
	7408	7496	7513	7519	7534	7558	7573	7578	7593	7615	7646	7652	7667	7742	7868
	7911	8003	8036	8062	8108	8124	8132	8147	8189	8246	8253	8268	8283	8326	8367
	8402	8420	8464	8531	8599	8667	8680	8830	8849	8876	11112	11119	11147		
POP	1919#	9292	9293	9705	9798	9995	10861	10880	10884	11048	11305	11694	11743	11775	11776
PUSH	1919#	9253	9255	9276	9694	9785	9969	10750	10761	11264	11668	11723	11756	11762	
REPORT	1919#														
RESDC	2713#	4744	4825	4894	4975	5044	5124	5195	5340	5646	5734	5817	5902	5987	6046
	6103	6161	6217	6273	6407	6482	6625	6750	6788	7419					
RKLOAD	2715#	4709	4732	4795	4814	4869	4883	4945	4962	5017	5031	5095	5110	5168	5182
	5305	5323	5371	5424	5524	5599	5618	5633	5683	5706	5721	5769	5793	5807	5854
	5874	5889	5939	5959	5974	6033	6090	6148	6204	6260	6304	6331	6454	6469	6546
	6597	6612	6673	6696	6737	6772	6826	6856	6899	6914	6956	6971	6991	7089	7104
	7126	7215	7232	7279	7310	7353	7404	7508	7530	7568	7589	7642	7663	7678	7690
	7734	7753	7768	7864	7900	7941	7956	8000	8032	8088	8104	8120	8143	8205	8242
	8264	8279	8294	8339	8362	8398	8416	8460	8511	8579	8664	8676	8818	8827	8839
	8846	8860	11108	11143											
SCOPE	1814#	2896	2927	2987	3196	3236	3286	3329	3373	3551	3599	3649	3715	3757	3818
	3881	3944	4086	4213	4289	4368	4420	4474	4534	4593	4690	4765	4846	4914	4995
	5065	5145	5214	5289	5358	5411	5472	5582	5666	5752	5837	5922	6009	6068	6125
	6182	6238	6291	6318	6353	6432	6506	6574	6653	6807	6882	6938	7071	7198	7260
	7462	7629	7720	7809	7847	7927	8019	8075	8228	8435	8468	8493	8562	8635	8799
	8974														
SETPRI	1919#	11572													
SETTRA	11816#	11825	11826	11827	11828	11830	11832	11833	11834	11835	11836	11837	11838	11839	11840
	11841	11842	11843	11844	11845	11846	11847	11848	11849	11850	11851	11852	11853	11854	11855
	11856	11857													
SETUP	1919#	2736													
SKIP	1919#														
SLASH	1919#														
SPACE	1919#														
STARS	1919#	2177	2179	2186	2199	2274	2277	2889	2895	2920	2926	2950	2986	3189	3195
	3226	3235	3278	3285	3322	3328	3355	3372	3541	3550	3588	3598	3636	3648	3707
	3714	3750	3756	3808	3817	3874	3880	3920	3943	4077	4085	4197	4212	4279	4288
	4359	4367	4412	4419	4464	4473	4525	4533	4584	4592	4681	4689	4755	4764	4836
	4845	4904	4913	4985	4994	5055	5064	5135	5144	5206	5213	5279	5288	5351	5357
	5404	5410	5458	5471	5574	5581	5657	5665	5744	5751	5828	5836	5913	5921	5998

J05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KC.P11 01-OCT-76 13:08

MACY11 27(1006) 05-OCT-76 09:17 PAGE 270
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0268

. ABS. 064604 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZR6KC.DSKZ:DZR6KC.SEQ/CRF/SOL/DOC=DZR6KC
RUN-TIME: 103 112 10 SECONDS
RUN-TIME RATIO: 398/226=1.7
CORE USED: 30K (59 PAGES)

DOCUMENT PAGES: 268