

RK611/RK06

FUNCTIONAL CONT DIAG
MD-11-DZR6K-D

EP-DZR6K-D-DL-C

APR 1977

COPYRIGHT © 1977



FICHE 1 OF 2

MADE IN USA

RK611/RK06

FUNCTIONAL CONT DIAG
MD-11-DZR6K-D

EP-DZR6K-D-DL-C

APR 1977

COPYRIGHT © 1977



FICHE 2 OF 2

MADE IN USA

The microfiche card contains a grid of frames. The left side of the card shows a series of frames, each containing technical diagrams and text. The right side of the card is mostly blank, with a few frames visible at the bottom right corner.

B01

EOF10ZR6J0SE0

00010000

770323

POP10 411

10MDR10ZR6K0SE0

00010000

770323

CO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 1
DZR6KD.P11 31-JAN-77 16:26

SEQ 0001

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZR6K-D-D
PRODUCT NAME: RK611 FUNCTIONAL CONTROLLER
DIAGNOSTIC
DATE: FEBRUARY, 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: MARV TEGROTENHUIS

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

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

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

COPYRIGHT (C) 1976 & 1977 BY DIGITAL EQUIPMENT CORPORATION

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

D01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 2
DZRGKD.P11 31-JAN-77 16:26

SEQ 0002

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

TABLE OF CONTENTS

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

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

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.

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.

H01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 6
 DZR6KD.P11 31-JAN-77 16:26

SEQ 0006

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

3.7 CONTROL Q (↑Q) OPERATION

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

3.8 UNIBUS ADDRESSES

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

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

3.9 EXECUTION TIME

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

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

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

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

4.0 PROGRAM DESCRIPTION

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

FIRST PASS - FIRST DRIVE:

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

FIRST PASS - ALL REMAINING DRIVES:

STATUS VALID TESTS UP TO THE MULTI-DRIVE OPERATIONS ARE PERFORMED ONCE ON EACH DRIVE.

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

SECOND AND ALL SUBSEQUENT PASSES:

THE SAME SEQUENCE OF TESTING IS REPEATED EXCEPT FOR TEST ITERATIONS WHICH ARE SPECIFIED FOR EACH TEST.

313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368

**BASIC INTERFACE AND OPTION TESTS

TEST 1 RK611 BASE ADDRESS TEST

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

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

**STATUS VALID TESTS

TEST 3 SELECT ALL DRIVES

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

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

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

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

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

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

NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE ANY OTHER TEST THAT FOLLOWS.

J01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 8
DZR6KD.P11 31-JAN-77 16:26

SEQ 0008

369
370

TEST 4 RELEASE ALL DRIVES

K01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 9
DZR6KD.P11 31-JAN-77 16:26

SEQ 0009

371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426

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 AVAILABLE 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 AVAILABLE 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

L01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 10
DZR6KD.P11 31-JAN-77 16:26

SEQ 0010

427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482

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 C. 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 SET. 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.

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

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 PACK 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

NO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 12
DZRB/D.P11 31-JAN-77 16:26

SEQ 0012

539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564

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.

565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619*****
**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

C02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 14
DZR6KD.P11 31-JAN-77 16:26

SEQ 0014

620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675

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 112
WORDS BEFORE THE LOCATION WITH BAD PARITY. MAKE SURE
THAT UNIBUS PARITY ERROR SETS.

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

**MULTIPLE SECTOR OPERATIONS

TEST 45 TWO SECTOR WRITE DATA (PART 1)

676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731

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

E02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 16
DZRBKD.P11 31-JAN-77 16:26

SEQ 0016

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,

732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787

F02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 17
DZR6KD.P11 31-JAN-77 16:26

SEQ 0017

788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843

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

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

TEST 64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

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

**SPECIAL DATA TRANSFER TESTS

TEST 65 MULTI-SECTOR DATA TRANSFER AND BSE

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:

G02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 18
 DZR6KD.P11 31-JAN-77 16:26

SEQ 0018

844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899

```
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 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

H02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 19
DZR6KD.P11 31-JAN-77 16:26

SEQ 0019

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 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.

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

900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955

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
AVAILABLE DRIVES.

TEST 104 FILE UNSAFE

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

TEST 105 DUMMY TEST FOR PREVIOUS TEST EXIT

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

IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE

956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011

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 SVXL 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:

1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066

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

NOTE1: IF SITEMB 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 ASCII DATA. EM IS ALWAYS A MESSAGE BUT DH CAN BE A MESSAGE OR A SET OF COLUMN LABELS SPACED ACROSS 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 ASCII 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

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

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 EXPECTED BUT NOT SET ERRORS AND THE UNEXPECTED BUT SET ERRORS.

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

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

PRECEDING 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.

1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234

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
RK3A 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

1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290

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

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

THE ERRORS REPORTED BY THIS FORMAT ARE:
NON-EXISTANT DRIVE
NON-EXISTANT MEMORY
CONTROLLER TIME OUT
UNIT FIELD ERROR
MULTIPLE DRIVE SELECT
PROGRAMMING ERROR
UNIBUS PARITY ERROR
ILLEGAL FUNCTION CODE

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

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

THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS. THE SPECIFIC ERRORS ARE:
WORD COUNT INCORRECT
BUS ADDRESS INCORRECT
CYLINDER ADDRESS INCORRECT
TRACK ADDRESS INCORRECT
SECTOR ADDRESS INCORRECT

ERROR ITEM 11
(ERROR INDICATOR OR STATUS BIT)
NOT SET AS A RESULT OF
(ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
TST NUM ERR PC DRIVE
\$TESTN \$ERRPC DRVNUM
RKCS1 RKCS2 RKDS RKER RKASOF RKMR1

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

STESTN SERRPC DRVNUM

GOOD BAD WORD NUM

SREG10 SREG11 SREG12

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 (TWTNN 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

1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346

1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402

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, TWAT15, TWAT25, TWAT85, 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.

D03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 28
DZR6KD.P11 31-JAN-77 16:26

SEQ 0028

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.CSI 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 TRANSISTIONS

1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458

E03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 29
DZR6KD.P11 31-JAN-77 16:26

SEQ 0029

.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,ROSTHD
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

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
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514

F03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 30
DZR6KD.P11 31-JAN-77 16:26

SEQ 0030

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

1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570

G03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 31
 DZR6KD.P11 31-JAN-77 16:26

SEQ 0031

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

1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626

H03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 32
DZR6KD.P11 31-JAN-77 16:26

SEQ 0032

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

BAD SECTOR CHECK

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

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

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).

1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738

- 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 RKMR1. 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
SSWR= 167400
STN= 1
.TITLE RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
; *COPYRIGHT (C) 1976 & 1977
; *DIGITAL EQUIPMENT CORP.
; *MAYNARD, MASS. 01754
; *
; *PROGRAM BY MARV TEGROTENHUIS
; *
; *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
; *PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
```

167400
000001

1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794

K03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 35
DZR6KD.P11 31-JAN-77 16:26

SEQ 0035

```

1795      ;*
1796      ;SBTTL  OPERATIONAL SWITCH SETTINGS
1797      ;*
1798      ;*      SWITCH      USE
1799      ;*      -----
1800      ;*      15      HALT ON ERROR
1801      ;*      14      LOOP ON TEST
1802      ;*      13      INHIBIT ERROR TYPEOUTS
1803      ;*      12      ABORT PROGRAM AFTER 20 ERRORS
1804      ;*      11      INHIBIT ITERATIONS
1805      ;*      10      BELL ON ERROR
1806      ;*      9      LOOP ON ERROR
1807      ;*      8      LOOP ON TEST IN SWR<7:0>
1808      ;SBTTL  BASIC DEFINITIONS
1809
1810      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
1811      ;STACK= 1100
1812      ;EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
1813      ;EQUIV  IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
1814
1815      ;*MISCELLANEOUS DEFINITIONS
1816      ;HT= 11      ;;CODE FOR HORIZONTAL TAB
1817      ;LF= 12      ;;CODE FOR LINE FEED
1818      ;CR= 15      ;;CODE FOR CARRIAGE RETURN
1819      ;CRLF= 200    ;;CODE FOR CARRIAGE RETURN-LINE FEED
1820      ;PS= 177776   ;;PROCESSOR STATUS WORD
1821      ;EQUIV  PS,PCU
1822      ;STKLMT= 177774 ;;STACK LIMIT REGISTER
1823      ;PIRQ= 177772  ;;PROGRAM INTERRUPT REQUEST REGISTER
1824      ;DSWR= 177570  ;;HARDWARE SWITCH REGISTER
1825      ;DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
1826
1827      ;*GENERAL PURPOSE REGISTER DEFINITIONS
1828      ;R0= %0      ;;GENERAL REGISTER
1829      ;R1= %1      ;;GENERAL REGISTER
1830      ;R2= %2      ;;GENERAL REGISTER
1831      ;R3= %3      ;;GENERAL REGISTER
1832      ;R4= %4      ;;GENERAL REGISTER
1833      ;R5= %5      ;;GENERAL REGISTER
1834      ;R6= %6      ;;GENERAL REGISTER
1835      ;R7= %7      ;;GENERAL REGISTER
1836      ;SP= %6      ;;STACK POINTER
1837      ;PC= %7      ;;PROGRAM COUNTER
1838
1839      ;*PRIORITY LEVEL DEFINITIONS
1840      ;PR0= 0      ;;PRIORITY LEVEL 0
1841      ;PR1= 40     ;;PRIORITY LEVEL 1
1842      ;PR2= 100    ;;PRIORITY LEVEL 2
1843      ;PR3= 140    ;;PRIORITY LEVEL 3
1844      ;PR4= 200    ;;PRIORITY LEVEL 4
1845      ;PR5= 240    ;;PRIORITY LEVEL 5
1846      ;PR6= 300    ;;PRIORITY LEVEL 6
1847      ;PR7= 340    ;;PRIORITY LEVEL 7
1848
1849      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1850      ;SW15= 100000

```

1851 040000
1852 020000
1853 010000
1854 004000
1855 002000
1856 001000
1857 000400
1858 000200
1859 000100
1860 000040
1861 000020
1862 000010
1863 000004
1864 000002
1865 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

1877 100000
1878 040000
1879 020000
1880 010000
1881 004000
1882 002000
1883 001000
1884 000400
1885 000200
1886 000100
1887 000040
1888 000020
1889 000010
1890 000004
1891 000002
1892 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

1905 000004
1906

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

1907	000010	RESVEC= 10	:: RESERVED AND ILLEGAL INSTRUCTIONS
1908	000014	TBITVEC=14	:: "T" BIT
1909	000014	TRIVEC= 14	:: TRACE TRAP
1910	000014	BPTVEC= 14	:: BREAKPOINT TRAP (BPT)
1911	000020	IOTVEC= 20	:: INPUT/OUTPUT TRAP (IOT) **SCOPE**
1912	000024	PLRVEC= 24	:: POWER FAIL
1913	000030	EMTVEC= 30	:: EMULATOR TRAP (EMT) **ERROR**
1914	000034	TRAPVEC=34	:: "TRAP" TRAP
1915	000060	TKVEC= 60	:: TTY KEYBOARD VECTOR
1916	000064	TPVEC= 64	:: TTY PRINTER VECTOR
1917	000240	PIRQVEC=240	:: PROGRAM INTERRUPT REQUEST VECTOR
1918		.SBTTL MEMORY MANAGEMENT DEFINITIONS	
1919		; *KT11 VECTOR ADDRESS	
1920			
1921			
1922	000250	MMVEC= 250	
1923			
1924		; *KT11 STATUS REGISTER ADDRESSES	
1925			
1926	177572	SR0= 177572	
1927	177574	SR1= 177574	
1928	177576	SR2= 177576	
1929	172516	SR3= 172516	
1930			
1931		; *KERNEL "I" PAGE DESCRIPTOR REGISTERS	
1932			
1933	172300	KIPDR0= 172300	
1934	172302	KIPDR1= 172302	
1935	172304	KIPDR2= 172304	
1936	172306	KIPDR3= 172306	
1937	172310	KIPDR4= 172310	
1938	172312	KIPDR5= 172312	
1939	172314	KIPDR6= 172314	
1940	172316	KIPDR7= 172316	
1941			
1942		; *KERNEL "I" PAGE ADDRESS REGISTERS	
1943			
1944	172340	KIPAR0= 172340	
1945	172342	KIPAR1= 172342	
1946	172344	KIPAR2= 172344	
1947	172346	KIPAR3= 172346	
1948	172350	KIPAR4= 172350	
1949	172352	KIPAR5= 172352	
1950	172354	KIPAR6= 172354	
1951	172356	KIPAR7= 172356	
1952			
1953	000210	AVECT1= 210	:: DEFINE RK611 VECTOR INTERRUPT
1954	000240	APRIOR= PR5	:: DEFINE RK611 PRIORITY
1955	177440	ABASE= 177440	:: DEFINE RK611 BASE BUS ADDRESS
1956			
1957		.SBTTL RK611 CONTROLLER REGISTER DEFINITION	
1958			
1959	000000	RKCS1= 0	:: CONTROL AND STATUS REGISTER 1
1960	000002	RKWC= 2	:: WORD COUNT REGISTER
1961	000004	RKBA= 4	:: BUS ADDRESS REGISTER
1962	000006	RKDA= 6	:: DESIRED TRACK SECTOR REGISTER

1963	000010	RKCS2= 10	: CONTROL AND STATUS REGISTER 2
1964	000012	RKDS= 12	: DRIVE STATUS REGISTER
1965	000014	RKER= 14	: ERROR REGISTER
1966	000016	RKASOF= 16	: ATTENTION SUMMARY AND OFFSET REGISTER
1967	000020	RKDCYL= 20	: DESIRED CYLINDER REGISTER
1968	000024	RKDB= 24	: DATA BUFFER
1969	000026	RKMR1= 26	: MAINTENANCE REGISTER 1
1970	000034	RKMR2= 34	: MAINTENANCE REGISTER 2
1971	000036	RKMR3= 36	: MAINTENANCE REGISTER 3
1972	000030	RKECPS= 30	: ECC POSITION INFORMATION
1973	000032	RKECPT= 32	: ECC PATTERN INFORMATION
1974	000022	RKSPAR= 22	: SPARE REGISTER
1975			
1976		.SBTTL DRIVE COMMANDS	
1977			
1978	000101	SELDRV= 101	: SELECT DRIVE
1979	000103	PACK= 103	: PACK ACKNOWLEDGE
1980	000105	CLEAR= 105	: DRIVE CLEAR
1981	000107	UNLOAD= 107	: UNLOAD
1982	000111	SRTSPL= 111	: START SPINDLE
1983	000113	RECAL= 113	: RECALIBRATE
1984	000115	OFFSET= 115	: OFFSET
1985	000117	SEEK= 117	: SEEK
1986	000121	RDATA= 121	: READ DATA
1987	000123	WRDATA= 123	: WRITE DATA
1988	000125	RDHEAD= 125	: READ HEADER
1989	000127	WRHEAD= 127	: WRITE HEADER AND DATA
1990	000131	WRTCHK= 131	: WRITE CHECK
1991	000300	INTR= 300	: GENERATE INTERRUPT TO CPU
1992			
1993		.SBTTL CONTROL AND STATUS REGISTER 1 BITS	
1994			
1995	000001	GO= BIT0	: GO BIT
1996	000100	IE= BIT6	: INTERRUPT ENABLE
1997	000200	RDY= BIT7	: CONTROLLER READY
1998	000400	BA16= BIT8	: BUS ADDRESS BIT 16
1999	001000	BA17= BIT9	: BUS ADDRESS BIT 17
2000	002000	CDT= BIT10	: CONTROLLER DRIVE TYPE (0=RK06)
2001	004000	CTO= BIT11	: CONTROLLER TIMED OUT WAITING FOR DRIVE RESPONSE
2002			
2003	010000	CFMT= BIT12	: CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
2004	020000	SPAR= BIT13	: DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
2005	040000	DI= BIT14	: DRIVE INTERRUPT
2006	100000	CERR= BIT15	: CONTROLLER ERROR
2007	100000	CCLR= BIT15	: CONTROLLER CLEAR
2008			
2009		.SBTTL CONTROL AND STATUS REGISTER 2 BITS	
2010			
2011	000007	DRVMSK= 7	: MASK FOR DRIVE SELECTION CODE
2012	000010	RLS= BIT3	: DESELECT OR RELEASE DRIVE IN BITS 0-2
2013	000020	BAI= BIT4	: BUS ADDRESS INCREMENT INHIBIT
2014	000040	SCLR= BIT5	: CLEAR CONTROLLER AND ALL DRIVES
2015	000100	IR= BIT6	: INPUT READY
2016	000200	OR= BIT7	: OUTPUT READY
2017	000400	UFE= BIT8	: UNIT FIELD ERROR
2018	001000	MDS= BIT9	: MULTIPLE DRIVE SELECT

B04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 39
CONTROL AND STATUS REGISTER 2 BITS

SEQ 0039

2019 002000
2020 004000
2021 010000
2022 020000
2023 040000
2024 100000

PGE= BIT10 ; PROGRAMMING ERROR
NEM= BIT11 ; NON-EXISTENT MEMORY
NED= BIT12 ; NON-EXISTENT DRIVE
UPE= BIT13 ; UNIT'S PARITY ERROR
WCE= BIT14 ; WRITE CHECK ERROR
DLT= BIT15 ; DATA LATE ERROR

.SBTTL ERROR REGISTER BIT DEFINITION

2028 000001
2029 000002
2030 000004
2031 000010
2032 000020
2033 000040
2034 000100
2035 000200
2036 000400
2037 001000
2038 002000
2039 004000
2040 010000
2041 020000
2042 040000
2043 100000

ILF= BIT0 ; ILLEGAL FUNCTION CODE
SKI= BIT1 ; SEEK INCOMPLETE
NXF= BIT2 ; NON-EXECUTABLE DRIVE FUNCTION
DRPAR= BIT3 ; DRIVE DETECTED DRIVE BUS PARITY ERROR
FMTE= BIT4 ; FORMAT ERROR
DTYE= BIT5 ; DRIVE TYPE ERROR
ECH= BIT6 ; ECC HARD
BSE= BIT7 ; BAD SECTOR ERROR
HVRC= BIT8 ; HEADER VRC ERROR
COE= BIT9 ; CYLINDER ADDRESS OVERFLOW ERROR
IDAE= BIT10 ; INVALID DISK ADDRESS ERROR
MLE= BIT11 ; WRITE LOCK ERROR
DTE= BIT12 ; DRIVE TIMING ERROR
OPT= BIT13 ; OPERATION (SEARCH) INCOMPLETE
UNS= BIT14 ; DRIVE UNSAFE
DCK= BIT15 ; DATA CHECK

.SBTTL STATUS REGISTER BIT DEFINITION

2046 000001
2047 000004
2048 000010
2049 000020
2050 000040
2051 000100
2052 000200
2053 000400
2054 001000
2055 002000
2056 004000
2057 020000
2058 040000
2059 100000

DRA= BIT0 ; DRIVE AVAILABLE (CONTROLLER IS SET IF
THIS BIT IS RESET)
OFST= BIT2 ; DRIVE OFFSET
ACLO= BIT3 ; AC LOW
SPDLSS= BIT4 ; SPEED LOSS
DROT= BIT5 ; DRIVE OFF TRACK
VV= BIT6 ; VOLUME VALID
DRDY= BIT7 ; DRIVE READY
DDT= BIT8 ; DRIVE TYPE (0=RK06)
WRL= BIT11 ; WRITE LOCK
PIP= BIT13 ; POSITIONING IN PROGRESS
DSC= BIT14 ; DRIVE STATUS CHANGE
SVAL= BIT15 ; STATUS VALID

.SBTTL MAINTENANCE REGISTER 1 BIT DEFINITION

2061 000017
2062 000017
2063 000017
2064 000017
2065 000020
2066 000040
2067 000100
2068 000200
2069 000400
2070 001000
2071 002000
2072 004000
2073 010000
2074 020000

MESMSK= 17 ; MESSAGE MASK
PAT= BIT4 ; FORCE EVEN PARITY ON SERCON MESSAGE LINES
DMO= BIT5 ; DIAGNOSTIC MODE
MSP= BIT6 ; MAINTENANCE SECTOR PULSE
MIND= BIT7 ; MAINTENANCE INDEX
MCLK= BIT8 ; MAINTENANCE CLOCK
MERD= BIT9 ; MAINTENANCE ENCODED READ DATA
MEWD= BIT10 ; MAINTENANCE ENCODED WRITE DATA
PCA= BIT11 ; PRECOMPENSATION ADVANCE
PCD= BIT12 ; PRECOMPENSATION DELAY
ECCW= BIT13 ; ECC WORD IS BEING READ OR WRITTEN

C04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
D7R6KD.P11 , 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 40
MAINTENANCE REGISTER 1 BIT DEFINITION

SEQ 0040

2075	040000	WRTGAT= BIT14	;WRITE GATE
2076	100000	RDGATE= BIT15	;READ GATE
2077		.SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A	
2078			
2079			
2080	000040	S.DRA= BIT5	;DRIVE AVAILIABLE
2081	000100	S.VV= BIT6	;VOLUME VALID
2082	000200	S.DRY= BIT7	;DRIVE READY
2083	000400	S.TYPE= BIT8	;DRIVE TYPE
2084	001000	S.FORM= BIT9	;DRIVE FORMAT
2085	002000	S.OFF= BIT10	;OFFSET
2086	004000	S.WRL= BIT11	;WRITE LOCK
2087	010000	S.SPIN= BIT12	;SPINDLE ON
2088	020000	S.PIP= BIT13	;POSITIONING IN PROGRESS
2089	040000	S.DSC= BIT14	;DRIVE STATUS CHANGE
2090			
2091		.SBTTL DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B	
2092			
2093	000040	S.ICYL= BIT5	;ILLEGAL CYLINDER ADDRESS
2094	000100	S.ACLO= BIT6	;AC LOW
2095	000200	S.FLT= BIT7	;DRIVE FAULT
2096	000400	S.ILF= BIT8	;ILLEGAL FUNCTION
2097	001000	S.PAR= BIT9	;DRIVE DETECTED SERCON PARITY ERROR
2098	002000	S.SKI= BIT10	;SEEK INCOMPLETE
2099	004000	S.WLE= BIT11	;WRITE LOCK ERROR
2100	010000	S.SPLS= BIT12	;SPEED LOSS
2101	020000	S.DROT= BIT13	;DRIVE OFF TRACK
2102	040000	S.UNS= BIT14	;DRIVE UNSAFE
2103			
2104		.SBTTL DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A	
2105			
2106	000020	S.XOOK= BIT4	;TRANSDUCER OK
2107	000040	S.HOHM= BIT5	;HEADS HOME
2108	000100	S.BRHM= BIT6	;BRUSHES HOME
2109	000200	S.DOOR= BIT7	;DOOR INTERLOCKED
2110	000400	S.CART= BIT8	;CARTRAGE INTERLOCK
2111	001000	S.SPOK= BIT9	;SPEED OK
2112	002000	S.FWD= BIT10	;FORWARD
2113	004000	S.REV= BIT11	;REVERSE
2114	010000	S.LOAD= BIT12	;HEADS LOADING
2115	020000	S.RTZ= BIT13	;RETURN TO ZERO
2116	040000	S.UNLD= BIT14	;HEADS UNLOADING
2117			
2118		.SBTTL DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B	
2119			
2120	000020	S.SECT= BIT4	;SECTOR ERROR
2121	000040	S.WCLK= BIT5	;WRITE CLOCK AND NO WRITE GATE
2122	000100	S.WGAT= BIT6	;WRITE GATE AND NO TRANSISTIONS
2123	000200	S.HOFL= BIT7	;HEAD FAULT
2124	000400	S.MHD= BIT8	;MULTIPLE HEAD SELECT
2125	001000	S.XERR= BIT9	;INDEX ERROR
2126	002000	S.DIB= BIT10	;DIBIT ERROR
2127	004000	S.PLO= BIT11	;PLO ERROR
2128	010000	S.NMOV= BIT12	;SEEK AND NO MOTION
2129	020000	S.LIMD= BIT13	;LIMIT DETECT ON SEEK
2130	040000	S.BRAKE= BIT14	;SERVO-BRAKE

D04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 41
DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B

SEQ 0041

```

2131
2132
2133
2134      000007      M.DRV= 7      ; DRIVE CODE
2135      100000      M.PAR= BIT15    ; PARITY
2136      000003      M.ID= 3         ; BYTE ID
2137      017760      M.COIF= 17760   ; CYLINDER DIFFERENCE/OFFSET
2138      017760      M.CADD= 17760   ; CYLINDER ADDRESS
2139      077770      M.SER= 77770    ; DRIVE SERIAL NUMBER
2140      000760      M.SECT= 760     ; SECTOR COUNT
2141      007000      M.HEAD= 7000    ; HEAD DECODE
2142
2143      .SBTTL  COMMON MASKS
2144
2145      000000      .=0
2146      ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
2147      ;*A ".+2" IOT SEQUENCE TO CATCH AND PROCESS ILLEGAL
2148      ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
2149      ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
2150      ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
2151      ;*LESS THAN 1002) JUMPS TO THE ERROR ROUTINE.
2152      ;*THE ERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
2153      ;*   PC=YYYYYY UNEXPECTED TRAP TO XXX
2154      ;*AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
2155      ;*WHERE XXX=LOCATION OF ILLEGAL TRAP
2156      ;*   YYYYYY=PC AT TIME OF TRAP
2157      ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
2158      ;*   CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.
2159      000000 000000 $40CAT: HALT      ; HALT
2160      000002 000737 BR      .-100    ; BRANCH TO 177700 & TIME OUT (NOT ON
2161      ;*   11/05)
2162      000004 001750 .WORD 3START    ; VECTOR TO STARTING ADDRESS
2163      000006 000340 .WORD 340      ; WITH PRIORITY LEVEL 7
2164      000174 000174 .=174
2165      000174 000000 DISPREG: .WORD 0 ; SOFTWARE DISPLAY REGISTER
2166      000176 000000 SWREG: .WORD 0  ; SOFTWARE SWITCH REGISTER
2167      .SBTTL  STARTING ADDRES(ES)
2168      000200 000137 001750 JMP      @START ;; GO TO START OF PROGRAM
2169      000204 000137 001740 JMP      RESTRT ; JUMP TO RESTART ROUTINE
2170      000214 000214 .=214
2171      000214 000137 001730 JMP      PARM ; JUMP TO OPERATOR ASSIGNED PARAMETERS
2172      000220 000220 .=220
2173      000220 000137 001720 JMP      SETCLK ; JUMP TO SET CLOCK ROUTINE
2174      .SBTTL  APT PARAMETER BLOCK
2175
2176      ;*****
2177      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
2178      ;*****
2179      000224      .SX=.      ; SAVE CURRENT LOCATION
2180      000024      .=24      ; SET POWER FAIL TO POINT TO START OF PROGRAM
2181      000024 200      FOR APT START UP
2182      000044      .=44      ; POINT TO APT INDIRECT ADDRESS PNTR.
2183      000044 $APTHDR ; POINT TO APT HEADER BLOCK
2184      000224      .=.SX     ; RESET LOCATION COUNTER
2185      ;*****
2186      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDPI1 DIAGNOSTIC

```

E04

RX611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 42
APT PARAMETER BLOCK

SEQ 0042

2187						
2188						
2189	000224		\$APTHD:			
2190	000224	000000	\$HIBTS: .WORD	0	:: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.	
2191	000226	001276	\$MBAOR: .WORD	\$MAIL	:: ADDRESS OF APT MAILBOX (BITS 0-15)	
2192	000230	000024	\$TSTM: .WORD	20.	:: RUN TIM OF LONGEST TEST	
2193	000232	000074	\$PASTM: .WORD	60.	:: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)	
2194	000234	000740	\$UNITM: .WORD	480.	:: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT	
2195	000236	000031		.WORD	\$ETEND-\$MAIL/2 ; ; LENGTH MAILBOX-ETABLE(WORDS)	

F04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 43
COMMON TAGS

SEQ 0043

.SBTTL COMMON TAGS

```

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

```

```

2196
2197
2198
2199
2200
2201
2202 001100
2203 001100
2204 001100 000000
2205 001102 000
2206 001103 000
2207 001104 000000
2208 001106 000000
2209 001110 000000
2210 001112 000000
2211 001114 000
2212 001115 001
2213 001116 000000
2214 001120 000000
2215 001122 000000
2216 001124 000000
2217 001126 000000
2218 001130 000000
2219 001132 000000
2220 001134 000
2221 001135 000
2222 001136 000000
2223 001140 177570
2224 001142 177570
2225 001144 177560
2226 001146 177562
2227 001150 177564
2228 001152 177566
2229 001154 000
2230 001155 002
2231 001156 012
2232 001157 000
2233 001160 000000
2234
2235 001162 000000
2236 001164 000000
2237 001166 000000
2238 001170 000000
2239 001172 000000
2240 001174 000000
2241 001176 000000
2242 001200 000000
2243 001202 000000
2244 001204 000000
2245 001206 000000
2246 001210 000000
2247 001212 000000
2248 001214 000000
2249 001216 000000
2250 001220 000000
2251 001222 000000

```

```

.=1100
SCMTAG: .WORD 0
STSTNM: .BYTE 00
SERFLG: .BYTE 00
SICNT: .WORD 00
SLPADR: .WORD 00
SLPERR: .WORD 00
SERTTL: .WORD 00
SITEMB: .BYTE 0
SERMAX: .BYTE 1
SERRPC: .WORD 00
SGDADR: .WORD 00
SBOADR: .WORD 00
SGDDAT: .WORD 00
SBDAT: .WORD 00
SAUTOB: .BYTE 00
SINTAG: .BYTE 00
SWR: .WORD 0SWR
DISPLAY: .WORD 0DISP
$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
$TMP0: .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

```

G04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 44
COMMON TAGS

SEQ 0044

2252	001224	000000	STMP1: .WORD	0	USER DEFINED
2253	001226	000000	STMP2: .WORD	0	USER DEFINED
2254	001230	000000	STMP3: .WORD	0	USER DEFINED
2255	001232	000000	STMP4: .WORD	0	USER DEFINED
2256	001234	000000	STMP5: .WORD	0	USER DEFINED
2257	001236	000000	STMP6: .WORD	0	USER DEFINED
2258	001240	000000	STMP7: .WORD	0	USER DEFINED
2259	001242	000000	STMP10: .WORD	0	USER DEFINED
2260	001244	000000	STMP11: .WORD	0	USER DEFINED
2261	001246	000000	STMP12: .WORD	0	USER DEFINED
2262	001250	000000	STMP13: .WORD	0	USER DEFINED
2263	001252	000000	STMP14: .WORD	0	USER DEFINED
2264	001254	000000	STMP15: .WORD	0	USER DEFINED
2265	001256	000000	STMP16: .WORD	0	USER DEFINED
2266	001260	000000	STMP17: .WORD	0	USER DEFINED
2267	001262	000000	STIMES: 0		MAX. NUMBER OF ITERATIONS
2268	001264	000000	SESCAPE: 0		ESCAPE ON ERROR ADDRESS
2269	001266	177607	SBELL: .ASCIZ	(207)<377><377>	CODE FOR BELL
2270	001272	077	SQUES: .ASCII	/?	QUESTION MARK
2271	001273	015	SCRLF: .ASCII	<15>	CARRIAGE RETURN
2272	001274	000012	SLF: .ASCIZ	<12>	LINE FEED
2273			;*****		
2274			.SBTTL APT MAILBOX-ETABLE		
2275			;*****		
2276			.EVEN		
2277			.MAIL:		
2278	001276		: APT MAILBOX		
2279	001276	000000	SMSGTY: .WORD	AMSGTY	: MESSAGE TYPE CODE
2280	001300	000000	SFATAL: .WORD	AFATAL	: FATAL ERROR NUMBER
2281	001302	000000	STESTN: .WORD	ATESTN	: TEST NUMBER
2282	001304	000000	SPASS: .WORD	APASS	: PASS COUNT
2283	001306	000000	SDEVCT: .WORD	ADEVCT	: DEVICE COUNT
2284	001310	000000	SUNIT: .WORD	AUNIT	: I/O UNIT NUMBER
2285	001312	000000	SMSGAD: .WORD	AMSGAD	: MESSAGE ADDRESS
2286	001314	000000	SMSGLG: .WORD	AMSGLG	: MESSAGE LENGTH
2287	001316		SETABLE:		: APT ENVIRONMENT TABLE
2288	001316	000	SENV: .BYTE	AENV	: ENVIRONMENT BYTE
2289	001317	000	SENVH: .BYTE	AENVH	: ENVIRONMENT MODE BITS
2290	001320	000000	SSWREG: .WORD	ASWREG	: APT SWITCH REGISTER
2291	001322	000000	SUSWR: .WORD	AUSWR	: USER SWITCHES
2292	001324	000000	SCPUOP: .WORD	ACPUOP	: CPU TYPE, OPTIONS
2293			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	001326	000	SMAMS1: .BYTE	AMAMS1	: HIGH ADDRESS, M.S. BYTE
2300	001327	000	SMTYP1: .BYTE	AMTYP1	: MEM. TYPE, BLK#1
2301			MEM. TYPE BYTE -- (HIGH BYTE)		
2302			900 NSEC CORE=001		
2303			300 NSEC BIPOLAR=002		
2304			500 NSEC MOS=003		
2305	001330	000000	SMADR1: .WORD	AMADR1	: HIGH ADDRESS, BLK#1
2306			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE		
2307	001332	000	SMAMS2: .BYTE	AMAMS2	: HIGH ADDRESS, M.S. BYTE

H04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR&KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 45
APT MAILBOX-ETABLE

SEQ 0045

2308 001333 000
2309 001334 000000
2310 001336 000
2311 001337 000
2312 001340 000000
2313 001342 000
2314 001343 000
2315 001344 000000
2316 001346 000210
2317 001350 000000
2318 001352 177440
2319 001354 000000
2320 001356 000000
2321 001360
2322

\$MTYP2: .BYTE AMTYP2 : MEM. TYPE, BLK#2
\$MAOR2: .WORD AMAOR2 : MEM. LAST ADDRESS, BLK#2
\$MAMS3: .BYTE AMAMS3 : HIGH ADDRESS, M.S. BYTE
\$MTYP3: .BYTE AMTYP3 : MEM. TYPE, BLK#3
\$MAOR3: .WORD AMAOR3 : MEM. LAST ADDRESS, BLK#3
\$MAMS4: .BYTE AMAMS4 : HIGH ADDRESS, M.S. BYTE
\$MTYP4: .BYTE AMTYP4 : MEM. TYPE, BLK#4
\$MAOR4: .WORD AMAOR4 : MEM. LAST ADDRESS, BLK#4
\$VECT1: .WORD AVECT1 : INTERRUPT VECTOR#1 BUS PRIORITY#1
\$VECT2: .WORD AVECT2 : INTERRUPT VECTOR#2 BUS PRIORITY#2
\$BASE: .WORD ABASE : BASE ADDRESS OF EQUIPMENT UNDER TEST
\$DEVN: .WORD ADEVN : DEVICE MAP
\$CDW1: .WORD ACDW1 : CONTROLLER DESCRIPTION WORD#1
\$ETEND:
.MEXIT

.SBTTL ERROR POINTER TABLE

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

```

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

```

.SERRTB:

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

```

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

```

```

EMIN: .WORD 0
      DH001
      DT001
      DF001

```

```

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

```

001360

```

001360 000000
001362 057134
001364 060506
001366 060600

```

J04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 47
ERROR POINTER TABLE

SEQ 0047

2379
2380
2381
2382
2383 001370 000000
2384 001372 000000
2385 001374 060514
2386 001376 060604
2387
2388
2389
2390
2391
2392
2393
2394 001400 000000
2395 001402 057162
2396 001404 060464
2397 001406 060624
2398
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

```

:*      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
      STSTN SERRPC  DRVNUM
      RKCS1  RKCS2  RKDS  RKER  RKASOF  RKMRI
      T.CS1  T.CS2  T.DS  T.ER  T.AST  T.MRI
EM3N:  .WORD  0
DH002A DT003
      DF003

```

```

* ERROR ITEMS 4,5,6,8,7 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 EXPECTED BUT NOT SET ERRORS
* AND THE UNEXPECTED BUT SET ERRORS.
*
* THE MESSAGE ITSELF EXPLAINS THE CIRCUMSTANCE FOR THE REPORT.
* INCLUDED IN THE REPORT WILL BE ONE OR MORE OF THE FOLLOWING
* STATEMENTS:
*
* "THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:"
* "THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:"
* "THE ABOVE ARE ERRORS SET IN OPERATION:"
*
* PRECEEDING ANY ONE OF THESE LINES WILL BE ONE OR MORE LINES THAT
* SPECIFY THE EXACT ERROR. FOLLOWING THE LAST LINE WILL BE A LINE
* THAT IDENTIFIES THE OPERATION BEING PERFORMED.
*
* EXAMPLE:
* NON-EXISTANT DRIVE
* THE ABOVE ARE ERRORS SET IN OPERATION:
* DRIVE SELECT
* (ADDITIONAL LINES OF INFORMATION)
*
* THIS IS THE RESULT OF AN ERROR SET IN A SELECT OPERATION.
*
* EXAMPLE:
* NON-EXISTANT DRIVE
* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
* DRIVE SELECT
* (ADDITIONAL LINES OF INFORMATION)
*
* THIS IS THE RESULT OF AN EXPECTED ERROR THAT DID NOT OCCUR, I.E.
* A NON-EXISTANT DRIVE WAS ADDRESSED BUT NED WAS NOT SET.
*

```

```

* EXAMPLE:
* NON-EXISTANT MEMORY
* THE ABOVE ARE UNEXPECTED ERRORS SET IN OPERATION:
* UNIBUS PARITY ERROR
* THE ABOVE ARE EXPECTED ERRORS THAT DID NOT SET IN OPERATION:
* WRITE DATA
* (ADDITIONAL LINES OF INFORMATION)
*
* THIS IS AN EXAMPLE OF NON-EXISTANT MEMORY BEING SET WHEN UNIBUS
* PARITY ERROR WAS EXPECTED.
*
* ERROR ITEM 4
* (DESCRIPTION OF ERROR)
* ERROR IN OPERATION
* (DESCRIPTION OF OPERATION)
* TST NUM ERR PC DRIVE
* STSTN SERRPC 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
* SREG10 SREG11 SREG12 SREG13 SREG14 SREG15 SREG16 SREG17
*
* 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

```

```

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.

```

```

EMS: .WORD 0
DHS: .WORD 0

```

```

2435
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 001410 000000
2476 001412 000000
2477 001414 060514
2478 001416 060634
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489 001420 000000
2490 001422 000000

```

L04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 49
ERROR POINTER TABLE

SEQ 0049

2491 001424 060514
2492 001426 060664
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514 001430 000000
2515 001432 000000
2516 001434 060514
2517 001436 060720
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540 001440 000000
2541 001442 000000
2542 001444 060514
2543 001446 060744
2544
2545
2546

DT005
DF005

*
* ERROR ITEM 6
* (DESCRIPTION OF ERROR)
* ERROR IN OPERATION
* (DESCRIPTION OF OPERATION)
* TST NUM ERR PC DRIVE
* STESTN SERRPC 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

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

*
* ERROR ITEM 7
* (DESCRIPTION OF ERR.
* ERROR IN OPERATION
* (DESCRIPTION OF OPERATION)
* TST NUM ERR PC DRIVE
* STESTN SERRPC 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
*
* DESCRIPTION OF OPERATION CAN BE ANY COMMAND, EITHER LEGAL OR ILLEGAL

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

*
* ERROR ITEM 10
* (DESCRIPTION OF ERROR)

M04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 50
ERROR POINTER TABLE

SEQ 0050

```

2547      *      ERROR AT COMPLETION OF OPERATION
2548      *      (DESCRIPTION OF OPERATION)
2549      *      TST NUM ERR PC DRIVE
2550      *      $TESTN $ERRPC DRVNUM
2551      *      EXPT IS
2552      *      $REG10 $REG11
2553      *
2554      *      THE ERRORS REPORTED BY THIS FORMAT ARE SOFTWARE DETECTED BY
2555      *      COMPARING EXPECTED RESULTS WITH ACTUAL RESULTS. THE SPECIFIC
2556      *      ERRORS ARE:
2557      *      WORD COUNT INCORRECT
2558      *      BUS ADDRESS INCORRECT
2559      *      CYLINDER ADDRESS INCORRECT
2560      *      TRACK ADDRESS INCORRECT
2561      *      SECTOR ADDRESS INCORRECT
2562
2563      001450 000000      EM10N: .WORD 0
2564      001452 057775      DM010
2565      001454 060564      DT015
2566      001456 060764      DF010
2567
2568      *      ERROR ITEM 11
2569      *      (ERROR INDICATOR OR STATUS BIT)
2570      *      NOT SET AS A RESULT OF
2571      *      (ANOTHER ERROR INDICATOR, STATUS BIT, OR OPERATION)
2572      *      TST NUM ERR PC DRIVE
2573      *      $TESTN $ERRPC DRVNUM
2574      *      RKCS1 RKCS2 RKDS RKER RKASOF RKMRI
2575      *      T.CS1 T.CS2 T.DS T.ER T.ASOF T.MRI
2576
2577      001460 000000      EM11N: .WORD 0
2578      001462 060121      DM011
2579      001464 060514      DT010
2580      001466 061004      DF011
2581
2582      *      ERROR ITEM 12
2583      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2584      *      "NOT RESET AS A RESULT OF"
2585
2586      001470 000000      EM12N: .WORD 0
2587      001472 060150      DM012
2588      001474 060514      DT010
2589      001476 061004      DF011
2590
2591      *      ERROR ITEM 13
2592      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2593      *      "SET AS A RESULT OF"
2594
2595      001500 000000      EM13N: .WORD 0
2596      001502 060201      DM013
2597      001504 060514      DT010
2598      001506 061004      DF011
2599
2600      *      ERROR ITEM 14
2601      *      THIS ERROR IS IDENTICAL TO ITEM 11 EXCEPT THE SECOND LINE IS:
2602      *      "RESET AS A RESULT OF"

```

N04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 51
ERROR POINTER TABLE

SEQ 0051

2603 001510 000000
2604 001512 060224
2605 001514 060514
2606 001516 061004
2607
2608
2609
2610
2611
2612
2613
2614
2615 001520 000000
2616 001522 057'62
2617 001524 060564
2618 001526 061026
2619
2620
2621
2622
2623 001530 000000
2624 001532 000000
2625 001534 060572
2626 001536 061036
2627

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

B05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 52
REGISTER STORAGE FOR TEST

SEQ 0052

.SBTTL REGISTER STORAGE FOR TEST

T.CS1:	.WORD	0
T.WC:	.WORD	0
T.BA:	.WORD	0
T.DA:	.WORD	0
T.CS2:	.WORD	0
T.DS:	.WORD	0
T.ER:	.WORD	0
T.ASOF:	.WORD	0
T.DCYL:	.WORD	0
T.SPAP:	.WORD	0
T.DB:	.WORD	0
T.MR1:	.WORD	0
T.ECPS:	.WORD	0
T.ECPT:	.WORD	0
T.MR2:	.WORD	0
T.MR3:	.WORD	0

.SBTTL REGISTER SETUP STORAGE

L.CS1:	.WORD	100
L.WC:	.WORD	0
L.BA:	.WORD	0
L.DA:	.WORD	0
L.DS:	.BYTE	0
L.DT:	.BYTE	0
L.CS2:	.WORD	0
L.ASOF:	.WORD	0
L.DCYL:	.WORD	0
L.MR1:	.WORD	0

;PRESET WITH INTERRUPT ENABLE

.SBTTL PROGRAM DEFINED VARIABLES

RKVEC:	.WORD	0
RKPRI:	.WORD	0
SRTFLG:	.WORD	0
DRVNUM:	.WORD	0
DRVBIT:	.WORD	0
ERRCNT:	.WORD	↑D20
ERRLMT:	.WORD	↑D20
BSF24P:	.WORD	BS24
BSF26P:	.WORD	BS26
BSS24P:	.WORD	0
BSS26P:	.WORD	0
BS26CT:	.WORD	0
BS24CT:	.WORD	0
MILCNT:	.WORD	↑D500
TIMCNT:	.WORD	↑D15
		.=1660

;RK VECTOR
;RK PRIORITY
;START FLAG
; 0 = 200
; 1 = 214
; -1 = 204
;DRIVE UNDER TEST
;WORD TO STORE BIT TO INDICATE DRIVE UNDER TEST
;ERROR COUNTER TO LIMIT ERROR
;ERRORS REPORTED IN PROGRAM
;DATA COMPARE ERROR LIMIT
;POINTER TO BAD SECTORS 24 SECTOR MODE
; (FACTORY)
;POINTER TO BAD SECTORS 26 SECTOR MODE
; (FACTORY)
;POINTER TO BAD SECTORS 24 SECT MODE
; (SOFTWARE)
;POINTER TO BAD SECTORS 26 SECTOR MODE
; (SOFTWARE)
;COUNT OF BAD SECTORS 26 SECTOR MODE
;COUNT OF BAD SECTORS 24 SECTOR MODE
;COUNT TO APPROXIMATE 1 MILL SEC
;COUNTER FOR MYTIME ROUTINE

2628	001540	000000
2629	001542	000000
2630	001544	000000
2631	001546	000000
2632	001548	000000
2633	001550	000000
2634	001552	000000
2635	001554	000000
2636	001556	000000
2637	001558	000000
2638	001560	000000
2639	001562	000000
2640	001564	000000
2641	001566	000000
2642	001570	000000
2643	001572	000000
2644	001574	000000
2645	001576	000000
2646		
2647		
2648	001600	000100
2649	001602	000000
2650	001604	000000
2651	001606	
2652	001606	000
2653	001607	000
2654	001610	000000
2655	001612	000000
2656	001614	000000
2657	001616	000000
2658		
2659		
2660	001620	000000
2661	001622	000000
2662	001624	000000
2663		
2664		
2665		
2666	001626	000000
2667	001630	000000
2668	001632	000024
2669		
2670	001634	000024
2671	001636	061166
2672		
2673	001640	061042
2674		
2675	001642	000000
2676		
2677	001644	000000
2678		
2679	001646	000000
2680	001650	000000
2681	001652	000764
2682	001654	000017
2683		001660

2684	001660	000000		
2685	001662	000000	INTSET: .WORD	0
2686				;NON-ZERO IF RK06 INTERRUPT SINCE
2687	001664	000000	OPTFLG: .WORD	0
2688				;LAST CLEARED
2689		000001		;OPTION FLAGS
2690		000002	DOTST= BIT0	;DRIVE 0 TO BE TESTED FLAG
2691		000004	MEMSZB= BIT1	;MEMORY SIZE REPORT FLAG
2692		000010	MEMPYB= BIT2	;MEMORY PARITY REPORT FLAG
2693		000100	SRTINS= BIT3	;START UP INSTRUCTIONS REPORTED FLAG
2694		000200	PARPRE= BIT6	;PARITY PRESENT FLAG
2695		000400	BSE RPT= BIT7	;BSE HAS BEEN REPORTED
2696		002000	FPFMT= BIT8	;FIRST PASS FORMAT SWITCH
2697		004000	CP1170= BIT10	;CP IS 11/70 FLAG
2698		100000	DRV RPT= BIT11	;DRIVE NUMBERS REPORTED FLAG
2699			LCLKPR= BIT15	;LINE CLOCK PRESENT
2700	001666	000000	DRVDRP: .WORD	0
2701	001670	000000	MEMPAR: .WORD	0
2702	001672	000000	CSRPTR: .WORD	0
2703				;LIST OF DRIVES DROPPED
2704	001674	000000	LCLKTK: .WORD	0
2705	001676	000000	REFMT: .WORD	0
2706				;WORD OF PARITY FLAGS
2707				;POINTER TO CSR TO USE FOR SETTING BAD
2708				;PARITY
2709	001700	000000		;LINE CLOCK TICK COUNTER
2710	001702	000000		;REFORMAT SWITCH FOR HALT
2711	001704	000000		
2712	001706	000000		
2713	001710	177546		
2714	001712	000100		
2715	001714	172100		
2716	001716	000114		

THE FOLLOWING 4 VARIABLES ARE USED TO STORE PARAMETERS FOR
HEADER OR DATA COMPARE CONTINUATION PROCESS.

2709	001700	000000	DESHLD: .WORD	0	;DESTINATION HOLD
2710	001702	000000	SRCHLD: .WORD	0	;SOURCE HOLD
2711	001704	000000	WRDNUM: .WORD	0	;WORD NUMBER IN ERROR HOLD
2712	001706	000000	WROCNT: .WORD	0	;WORDS LEFT IN COMPARE HOLD
2713	001710	177546	KW11ADD: .WORD	177546	;KW11-L ADDRESS
2714	001712	000100	KW11VEC: .WORD	100	;KW11-L VECTOR
2715	001714	172100	MM11BAS: .WORD	172100	;MM11 ADDRESS
2716	001716	000114	MM11VECA: .WORD	114	;MM11 VECTOR

2773	002204	012716	002212	64S:	MOV	#65S,(SP)	;;SET UP FOR TRAP RETURN
2774	002210	000002			RTI		
2775	002212	012737	000176 001140	65S:	MOV	#SWREG,SWR	;;POINT TO SOFTWARE SWR
2776	002220	012737	000174 001142		MOV	#DISPREG,DISPLAY	
2777	002226	012637	000004	66S:	MOV	(SP)+,2#ERRVEC	;;RESTORE ERROR VECTOR
2778							
2779	002232	005037	001304		CLR	\$PASS	;;CLEAR PASS COUNT
2780	002236	132737	000200 001317		BITB	#APTSIZE,\$ENVM	;;TEST USER SIZE UNDER APT
2781	002244	001403			BEQ	67S	;;YES,USE NON-APT SWITCH
2782	002246	012737	001320 001140		MOV	#SSWREG,SWR	;;NO,USE APT SWITCH REGISTER
2783	002254			67S:			
2784				.SBTTL	TYPE PROGRAM NAME		
2785				;;TYPE	THE NAME OF THE PROGRAM IF FIRST PASS		
2786	002254	005227	177777		INC	#-1	;;FIRST TIME?
2787	002260	001066			BNE	68S	;;BRANCH IF NO
2788	002262	022737	032160 000042		CMP	#SENDAD,2#42	;;ACT-11?
2789	002270	001462			BEQ	68S	;;BRANCH IF YES
2790	002272	104401	002340		TYPE	69S	;;TYPE ASCIZ STRING
2791				.SBTTL	GET VALUE FOR SOFTWARE SWITCH REGISTER		
2792	002276	005737	000042		TST	2#42	;;ARE WE RUNNING UNDER XXDP/ACT?
2793	002302	001012			BNE	70S	;;BRANCH IF YES
2794	002304	123727	001316 000001		CMPB	\$ENV, #1	;;ARE WE RUNNING UNDER APT?
2795	002312	001406			BEQ	70S	;;BRANCH IF YES
2796	002314	023727	001140 000176		CMP	SWR, #SWREG	;;SOFTWARE SWITCH REG SELECTED?
2797	002322	001005			BNE	71S	;;BRANCH IF NO
2798	002324	104406			GTSWR		;;GET SOFT-SWR SETTINGS
2799	002326	000403			BR	71S	
2800	002330	112737	000001 001134	70S:	MOVB	#1,\$AUTOB	;;SET AUTO-MODE INDICATOR
2801	002336			71S:			
2802	002336	000437			BR	68S	;;GET OVER THE ASCIZ
2803				;;69S:	.ASCIZ	<CRLF>*RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MAINDEC-11-DZR6K-D*<CRLF>	
2804	002436			68S:			
2805	002436	132737	000200 001317		BITB	#BIT7,\$ENVM	;;TEST IF DO NOT SIZE
2806	002444	001043			BNE	3S	;;YES - SKIP
2807	002446	004737	032200		JSR	PC,\$SIZE	
2808	002452	023727	032504 000740		CMP	\$LSTBK, #740	;;MAKE SURE MEMORY IS SUFFICIENT
2809	002460	103007			BHIS	2S	;;YES - SKIP
2810	002462	104401	050555		TYPE	OPROOS	;;MESSAGE (NOT ENOUGH MEMORY)
2811	002466	012737	000001 032016		MOV	#1,\$EOPCT	;;FORCE END OF PROGRAM
2812	002474	000137	031770		JMP	\$EOP	
2813	002500	013700	032504	2S:	MOV	\$LSTBK,R0	;;GET LAST BANK
2814	002504	012701	000006		MOV	#6,R1	;;SET SHIFT COUNT
2815	002510	013703	032502		MOV	\$LSTAD,R3	;;GET LAST ADDRESS
2816	002514	005004			CLR	R4	;;CLEAR R4 FOR OVERFLOW
2817	002516	005737	032236		TST	\$KT11	;;MEM MANAGE PRESENT?
2818	002522	100005			BPL	23S	;;NO - SKIP
2819	002524	006100		22S:	ROL	R0	;;SHIFT BANK LEFT
2820	002526	006104			ROL	R4	;;ADD IN CARRY
2821	002530	005301			DEC	R1	;;DECREMENT COUNT
2822	002532	001374			BNE	22S	;;LOOP IF NOT ZERO
2823	002534	050003			BIS	R0,R3	;;SET BANK BITS IN LAST ADDRESS
2824	002536	112737	000001 001327	23S:	MOVB	#1,\$MTYP1	;;FORCE MEMORY TYPE TO 1
2825	002544	110437	001326		MOVB	R4,\$MAMS1	;;STORE UPPER MEMORY ADDRESS
2826	002550	010337	001330		MOV	R3,\$MADR1	;;STORE LOWER ADDRESS
2827	002554	032737	000010 001664	3S:	BIT	#SATINS,OPTFLG	;;TEST IF ALREDY REPORTED
2828	002562	001005			BNE	24S	;;YES - SKIP

F05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 56
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0056

2829	002564	104401	051664		TYPE	OPR016	;TYPE STARTUP INSTRUCTIONS
2830	002570	052737	000010	001664	BIS	%SRATNS,OPTFLG	;SET REPORTED FLAG
2831	002576				24\$:		
2832	002576	022737	000001	001624	CMP	%1,SRATFLG	;CHECK IF PARAMETER START
2833	002604	001122			BNE	15\$	;NO START TESTING
2834	002606	104401	050446		5\$:	TYPE OPR001	;TYPE "RK611 BUS ADDRESS () ="
2835	002612	013746	001352		MOV	%BASE,-(SP)	;SAVE %BASE FOR TYPEOUT
2836	002616	104402			TYPOC		;GO TYPE--OCTAL ASCII(ALL DIGITS)
2837	002620	104401	050475		TYPE	,OPR002	
2838	002624	104412			RDOCT		;GET VALUE
2839	002626	012637	001222		MOV	(SP)+,%STMP0	
2840	002632	001407			BEQ	7\$	;CHECK IF <CR>
2841	002634	022737	160000	001222	CMP	%160000,%STMP0	;CHECK IF IN I/O PAGE
2842	002642	101361			BHI	5\$	
2843	002644	013737	001222	001352	MOV	%STMP0,%BASE	;LOAD NEW BUS ADDRESS
2844	002652	104401	050503		7\$:	TYPE OPR003	;TYPE "RK611 VECTOR ADDRESS () ="
2845	002656	013746	001346		MOV	%VECT1,-(SP)	;GET %VECT1 FOR TYPEOUT
2846	002662	042716	160000		BIC	%160000,(SP)	;CLEAR PRIORITY BITS
2847	002666	104402			TYPOC		
2848	002670	104401	050475		TYPE	,OPR002	
2849	002674	104412			RDOCT		;GET VALUE
2850	002676	012637	001222		MOV	(SP)+,%STMP0	
2851	002702	001412			BEQ	10\$	;CHECK IF <CR>
2852	002704	022737	001000	001222	CMP	%1000,%STMP0	
2853	002712	101757			BLOS	7\$	;CHECK IF LEGAL
2854	002714	042737	017777	001346	BIC	%17777,%VECT1	;CLEAR OLD VECTOR
2855	002722	053737	001222	001346	BIS	%STMP0,%VECT1	;LOAD NEW VECTOR ADDRESS
2856	002730	104401	050533		10\$:	TYPE OPR004	;TYPE "RK611 PRIORITY () ="
2857	002734	005046			CLR	-(SP)	
2858	002736	113716	001347		MOVB	%VECT1+1,(SP)	
2859	002742	006216			ASR	(SP)	;SHIFT 5 BITS RIGHT
2860	002744	006216			ASR	(SP)	
2861	002746	006216			ASR	(SP)	
2862	002750	006216			ASR	(SP)	
2863	002752	006216			ASR	(SP)	
2864	002754	104402			TYPOC		
2865	002756	104401	050475		TYPE	,OPR002	
2866	002762	104412			RDOCT		;GET VALUE
2867	002764	012637	001222		MOV	(SP)+,%STMP0	
2868	002770	001430			BEQ	15\$	;CHECK IF <CR>
2869	002772	022737	000007	001222	CMP	%7,%STMP0	;CHECK IF LEGAL
2870	003000	103753			BLO	10\$	
2871	003002	022737	000004	001222	CMP	%4,%STMP0	
2872	003010	101347			BHI	10\$	
2873	003012	006337	001222		ASL	%STMP0	;SHIFT 5 BITS LEFT
2874	003016	006337	001222		ASL	%STMP0	
2875	003022	006337	001222		ASL	%STMP0	
2876	003026	006337	001222		ASL	%STMP0	
2877	003032	006337	001222		ASL	%STMP0	
2878	003036	042737	160000	001347	BIC	%160000,%VECT1+1	;CLEAR OLD PRIORITY
2879	003044	053737	001222	001347	BIS	%STMP0,%VECT1+1	;LOAD RK611 PRIORITY
2880	003052	005037	001304		15\$:	CLR %PASS	;SET PASS COUNT TO ZERO
2881	003056	005037	001666		CLR	DRVDRP	;CLEAR DROPPED DRIVES LIST
2882	003062	042737	004000	001664	BIC	%DRVDRPT,OPTFLG	;CLEAR DRIVE #'S REPORTED FLAG
2883	003070	004737	034434		JSR	PC,OPTTST	;SETUP PARITY CHECK & CLOCK
2884	003074	013700	001346		MOV	%VECT1,RO	;STORE VECTOR FOR USE

G05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 57
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0057

2885	003100	042700	160000		BIC	#160000,R0	;CLEAR PRIORITY BITS
2886	003104	010037	001620		MOV	R0,RKVEC	
2887	003110	012710	034354		MOV	#INTHLR,(R0)	;SETUP INTERRUPT ADDRESS
2888	003114	113737	001347	001622	MOVB	\$VECT1+1,RKPRI	;STORE PRIORITY FOR USE
2889	003122	013702	001352		MOV	\$BASE,R2	;SET BASE ADDRESS
2890	003126	005037	001264		CLR	\$ESCAPE	;CLEAR ESCAPE
2891	003132	012746	000000		MOV	#PRO,-(SP)	;SET PRIORITY
2892	003136	012746	003144		MOV	#16\$,-(SP)	
2893	003142	000002			RTI		
2894	003144						
2895							

16\$:

H05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 58
**BASIC INTERFACE AND OPTION TESTS

SEQ 0058

```

2896
2897
2898
2899
2900
2901
2902
2903
2904 003144 000004
2905 003146 012737 000100 001262
2906 003154 012706 001100
2907 003160 012701 000004
2908 003164 012146
2909 003166 011146
2910 003170 012701 000004
2911 003174 012721 034346
2912 003200 012711 000340
2913 003204 013702 001352
2914 003210 005037 001662
2915 003214 012762 000000 000000
2916 003222 000240
2917 003224 000240
2918 003226 000240
2919 003230 005737 001662
2920 003234 001406
2921 003236 012737 053303 001360
2922 003244 104001
2923 003246 000137 043404
2924 003252 012701 000006
2925 003256 012611
2926 003260 012641
2927
2928
2929
2930
2931
2932
2933
2934
2935 003262 000004
2936 003264 012737 000100 001262
2937 003272 012762 005000 000010
2938
2939 003300 005037 001662
2940 003304 012762 000300 000000
2941 003312 000240
2942 003314 000240
2943 003316 000240
2944 003320 005737 001662
2945 003324 001011
2946 003326 105737 001103
2947
2948
2949 003332 001004
2950 003334 012737 053303 001360
2951 003342 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.
*****
*ST1: 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 $BASE,R2 ;SET POINTER TO RK611 BASE ADDRESS
CLR INTSET ;CLEAR INTERRUPT COUNTER
MOV #0,RKCS1(R2) ;WRITE CSI TO SEE IN NEM WILL SET
NOP
NOP
NOP
TST INTSET ;TEST IF COUNTER IS 0
BEG 1$ ;YES - SKIP ERROR REPORT
MOV #EM1,EMIN ;MESSAGE (NON-EXISTANT MEMORY TRAP ERR)
ERROR 1
JMP CTRHLT ;GO TO CONTROLLED HALT
1$: MOV #6,R1 ;RESTORE VECTOR
MOV (SP)+,(R1)
MOV (SP)+,-(R1)
*****
*TEST 2 INTERRUPT VECTOR ADDRESS TEST
* CHECK THAT THE INTERRUPT VECTOR FOR THE RK611 IS SET TO THE
* EXPECTED ADDRESS. IF INTERRUPT VECTOR IS IN ERROR,
* THE PROGRAM IS HALTED.
*****
*ST2: SCOPE
MOV #100,$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 CSI 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

```

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 59
T2 INTERRUPT VECTOR ADDRESS TEST

SEQ 0059

2952 003344 000137 043404
2953 003350

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

.SBTTL **STATUS VALID TESTS

TEST 3 SELECT ALL DRIVES

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

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

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

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

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

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

NOTE: THIS TEST MUST BE RUN AT LEAST ONCE BEFORE
ANY OTHER TEST THAT FOLLOWS.

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 #IS, -(SP) ;RK611 INTERRUPTS
RTI

1\$: MOV RKVEC, R1 ;GET VECTOR
MOV #INTHLR, (R1)+ ;LOAD INTERRUPT VECTOR

2997 003350 000004
2998 003352 012737 000062 001262
2999 003360 104416
3000 003362 104003
3001
3002 003364 012746 000000
3003 003370 012746 003376
3004 003374 000002
3005
3006 003376 013701 001620
3007 003402 012721 034354

J05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 60
T3 SELECT ALL DRIVES

SEQ 0060

3008	003406	012711	000340		MOV	#PR7,(R1)	
3009	003412	012703	001354		MOV	#SDEV,R3	; GET ADDRESS OF DEVICE MAP
3010	003416	132737	000200	001317	BITB	#BIT7,\$ENVN	; TEST IF SHOULD SIZE
3011	003424	001007			BNE	50\$	; NO - SKIP DRIVE SIZING.
3012	003426	005737	001304		TST	\$PASS	; TEST IF FIRST PASS
3013	003432	001004			BNE	50\$	; NO - SKIP TO DRIVE SELECT TEST
3014	003434	032737	004000	001664	BIT	#DRV PT,OPTFLG	; TEST IF DRIVE #'S REPORTED
3015	003442	001402			BEQ	92\$	; NO - GOTO SIZING
3016	003444	000137	004032		JMP	11\$	
3017	003450	003013			CLR	(R3)	; CLEAR DEVICE MAP
3018	003452	123727	000041	000013	CMPB	#41,#13	; TEST IF RK06 IS LOAD DEVICE
3019	003460	001077			BNE	77\$	; NO - SKIP
3020	003462	022737	000001	001624	CMP	#1,\$RTFLG	; WAS START AT PARAM?
3021	003470	001411			BEQ	2\$	; YES - SKIP
3022	003472	104401	050776		TYPE	,OPR007	; NO TEST OF DRIVE 0
3023	003476	052737	000001	001666	BIS	#BIT0,DRVDRP	; SET DRIVE 0 DROPPED
3024	003504	042737	000001	001664	BIC	#DOTST,OPTFLG	; DR FLAG - NO TEST DRIVE 0
3025	003512	000465			BR	7\$	
3026	003514	104401	050626		TYPE	,OPR006	; MESSAGE - SWAP PACK ON DRIVE OFF LINE.
3027	003520	005037	001610		CLR	,LCS2	; SET TO DRIVE 0
3028	003524	005037	001232		CLR	\$TMP4	; CLEAR FOR USE AS A SWITCH
3029	003530	012737	000101	001600	MOV	#SELDLV,LCS1	; LOAD FOR SELECT
3030	003536	012737	003600	001264	MOV	#4\$,\$ESCAPE	; SET UP ESCAPE IN CASE OF ERR
3031	003544	104417			TLOADRK		; LOAD RK & DO SELECT
3032							
3033	003546	104423			THAT16		; WAIT 16MS FOR COMPLETION
3034	003550	104002			ERROR	2	; NOT DONE ON TIME
3035							
3036	003552	104420			TGETRK		; GET RK REGISTER
3037	003554	032737	100000	001540	BIT	#CERR,T.CS1	; TEST IF CERR
3038	003562	001417			BEQ	5\$	; NO - SKIP
3039	003564	032737	010000	001550	BIT	#NED,T.CS2	; TEST IF NED
3040	003572	001002			BNE	4\$	; YES - SKIP
3041							
3042	003574	104421			TCHKOP		; CHECK THE OPERATION AND REPORT THE ERROR
3043	003576	104004			ERROR	4 ;OR5,6,7	; AFTER THE ERROR IS REPORTED THE TEST
3044							; IS ABORTED
3045	003600	104401	050776		TYPE	,OPR007	; TYPE NO TEST OF DRIVE 0
3046	003604	042737	000001	001664	BIC	#DOTST,OPTFLG	; DR FLAG - NO TEST OF DRIVE 0
3047	003612	052737	000001	001666	BIS	#BIT0,DRVDRP	; SET DRV 0 AS DROPPED
3048	003620	000422			BR	7\$	; SKIP OVER WAIT FOR PACK MOUNT
3049	003622	005737	001232		TST	\$TMP4	; TEST FLAG DRIVE READY HAS RESET
3050	003626	001010			BNE	6\$	; YES - SKIP TO CHECK IF IT IS SET AGAIN
3051	003630	032737	000200	001552	BIT	#DRDY,T.DS	; ELSE CHECK READY
3052	003636	001334			BNE	3\$	; STILL SET - GET STATUS AGAIN
3053	003640	012737	177777	001232	MOV	#-1,\$TMP4	; ELSE SET FLAG TO INDICATE READY WENT LOW
3054	003646	000730			BR	3\$	; GO GET STATUS AGAIN
3055							
3056	003650	032737	000200	001552	BIT	#DRDY,T.DS	; TEST IF READY SET AGAIN
3057	003656	001724			BEQ	3\$	; NO - GO GET STATUS AGAIN
3058	003660	052737	000001	001664	BIS	#DOTST,OPTFLG	; ELSE SET DRV 0 TEST FLAG.
3059							
3060	003666	013737	003770	001264	MOV	35\$,\$ESCAPE	; SET UP ESCAPE IN CASE OF ERR
3061	003674	005000			CLR	R0	; CLEAR FOR DRIVE NUMBER COUNTER
3062	003676	012701	000001		MOV	#1,R1	; SET BIT 0 AS DRIVE SELECTOR
3063							

K05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 61
T3 SELECT ALL DRIVES

SEQ 0061

3064	003702	032737	000001	001664	BIT	#DOTST,OPTFLG	;TEST DRIVE 0?
3065	003710	001433			BEQ	9\$	;NO - SKIP
3066							
3067	003712	104416			8\$: TSSINIT		;INITIALIZE SUBSYSTEM
3068	003714	104003			ERROR	3	;ERROR IF NOT SUCCESSFUL
3069							
3070	003716	010037	001610		MOV	RO,L.CS2	;LOAD DRIVE NUMBER
3071	003722	012737	000.01	001600	MOV	#SELDIV,L.CS1	;LOAD DRIVE SELECT
3072	003730	104417			TLOADRK		;LOAD RK REGS
3073							
3074	003732	104423			TWAT16		;WAIT FOR INTERRUPT
3075	003734	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
3076							
3077	003736	104420			TGETRK		;GET RK REGS
3078	003740	032737	100000	001540	BIT	#CERR,T.CS1	;ERROR?
3079	003746	001022			BNE	10\$	;YES - SKIP
3080	003750	032737	000200	001552	BIT	#DRDY,T.DS	;ELSE TEST IF DRIVE READY
3081	003756	001404			BEQ	35\$	;NO - SKIP
3082	003760	032737	004000	001552	BIT	#WRL,T.DS	;ELSE TEST IF WRITE LOCKED
3083	003766	001403			BEQ	36\$	;NO - SKIP
3084	003770	050137	001666		35\$: BIS	R1,DRVDRP	;SET THIS BIT AS DROPPED DRIVE
3085	003774	000401			BR	9\$	
3086	003776				36\$: BIS		
3087	003776	050113			BIS	R1,(R3)	;SET BIT - DRIVE PRESET IN MAP
3088							
3089	004000	005200			9\$: INC	RO	;BUMPS TO NEXT DRIVE
3090	004002	006301			ASL	R1	;SHIFT DRIVE SELECTOR TO NEXT DRIVE.
3091	004004	032701	000400		BIT	#BIT8,R1	;WAS LAST DRIVE DONE?
3092	004010	001740			BEQ	8\$	;YES - SKIP
3093	004012	000407			BR	11\$	;ELSE LOOP TO SELECT NEXT DRIVE
3094							
3095	004014	032737	010000	001550	10\$: BIT	#NED,T.CS2	;WAS CERR DUE TO NED?
3096	004022	001366			BNE	9\$	;YES - BUMP TO NEXT DRIVE
3097							
3098	004024	104421			TCHKOP		;ELSE REPORT THE ERRORS
3099	004026	104004			ERROR	4 ;OR5,6,7	
3100	004030	000000			.WORD	0	
3101	004032	032737	004000	001664	101\$: BIT	#DRVPT,OPTFLG	;TEST IF SHOULD REPORT
3102	004040	001037			BNE	16\$	;NO - SKIP
3103							
3104	004042	005713			TST	(R3)	;ANY DRIVE AVAILABLE?
3105	004044	001004			BNE	12\$	;BR IF NOT ZERO
3106	004046	104401	051141		TYPE	OPR008	;ELSE REPORT NO DRIVES AVAILABLE
3107	004052	000137	043404		JMP	CTRHLT	;GO TO CONTROLLED HALT
3108							
3109	004056	012701	000200		12\$: MOV	#BIT7,R1	;SET DRIVE SELECTOR FOR DRIVE 7
3110	004062	012700	000007		MOV	#7,RO	;SET DRIVE NUMBER TO 7
3111	004066	104401	051224		TYPE	,OPR009	;TYPE PREFIX TO DRIVE TEST MESSAGE
3112							
3113	004072	030113			13\$: BIT	R1,(R3)	;TEST IF THIS DRIVE TO BE TESTED
3114	004074	001007			BNE	15\$	;YES - SKIP
3115							
3116	004076	005300			14\$: DEC	RO	;ELSE DECREMENT DRIVE NUMBER
3117	004100	006201			ASR	R1	;SHIFT BIT SELECTOR TO NEXT DRIVE DOWN
3118	004102	001373			BNE	13\$	;IF NOT SHIFTED OUT - LOOP
3119	004104	052737	004000	001664	BIS	#DRVPT,OPTFLG	;SET DRIVE #'S REPORTED FLAG

L05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.F11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 62
T3 SELECT ALL DRIVES

SEQ 0062

3120	004112	000412			BR	16\$	;ELSE GO TO STATUS VALID TEST
3121							
3122	004114	010037	004030		15\$: MOV	RO,101\$	;PUT DRIVE NUMBER IN TYPE LOCATION
3123	004120	052737	000060	004030	BIS	#BIT4:BITS,101\$	;MAKE IT ASCII
3124	004126	104401			TYPE		;TYPE IT
3125	004130	004030			101\$		
3126	004132	104401	050443		TYPE	SPACE2	;TYPE SOME SPACES
3127	004136	000757			BR	14\$	;LOOP
3128							
3129	004140	005000			16\$: CLR	RO	;CLEAR DRIVE NUMBER COUNTER
3130	004142	012701	000001		MOV	#1,R1	;SET DRIVE SELECTOR TO DRIVE 0
3131	004146	012737	177777	001240	MOV	#-1,\$TMP7	;SET \$TMP7 NEGATIVE
3132	004154	012737	177777	001630	MOV	#-1,DRVBIT	;SET DRIVE SELECT BIT NEGATIVE
3133	004162	012737	004264	001264	MOV	#18\$,SESCAPE	;SET ESCAPE IN CASE OF ERR
3134	004170	030137	001666		BIT	R1,DRVDRP	;HAS DRIVE 0 BEEN DROPPED?
3135	004174	001033			BNE	18\$	;YES - SKIP TO DRIVE INC.
3136							
3137	004176	104416			17\$: TSSINIT		;CLEAR SUBSYSTEM
3138	004200	104003			ERROR	3	;ERROR FOR BAD CLEAR
3139							
3140	004202	010037	001610		MOV	RO,L.CS2	;SET DRIVE SELECT
3141	004206	010037	001626		MOV	RO,DRVNUM	;SET DRIVE NUMBER
3142	004212	012737	000101	001600	MOV	#SELDRV,L.CS1	;SET FOR DRIVE SELECT
3143							
3144	004220	104417			TLOADRK		;LOAD RK REGS
3145	004222	104423			THAT16		;WAIT FOR INTERRUPT
3146	004224	104002			ERROR	2	;ERROR TO SLOW/NOT COMPLETE
3147	004226	030113			BIT	R1,(R3)	;WAS THAT DRIVE AVAILABLE
3148	004230	001026			BNE	19\$	;YES - SKIP
3149							
3150	004232	104422			TCHKWE		;CHECK THAT ERROR OCCURRED
3151	004234	000000			.WORD	0	;NONE OF GROUP 1
3152	004236	000000			.WORD	0	;NONE OF GROUP 2
3153	004240	000001			.WORD	1	;NED IN GROUP 3
3154	004242	104004			ERROR	4 ;OR5,6,7	;ERROR IF NO ERROR OR WRONG ERROR
3155							
3156	004244	032737	100000	001552	BIT	#SVAL,T.DS	;DID STATUS VALID RESET?
3157	004252	001404			BEQ	18\$	;YES - SKIP
3158	004254	012737	055425	001400	MOV	#EM47,EM3N	;ELSE MESSAGE (SVAL NOT RESET WITH NED)
3159	004262	104003			ERROR	3	
3160	004264	005200			18\$: INC	RO	;BUMP TO NEXT DRIVE
3161	004266	001301			ASL	R1	;SHIFT DRIVE SELECT BIT
3162	004270	032701	000400		BIT	#BIT8,R1	;ALL DRIVES CHECKED
3163	004274	001063			BNE	21\$	;YES - EXIT
3164	004276	030137	001666		BIT	R1,DRVDRP	;TEST IF DRIVE HAS BEEN DROPPED
3165	004302	001370			BNE	18\$	;YES - GET NEXT DRIVE
3166	004304	000734			BR	17\$	;ELSE CHECK THIS DRIVE
3167	004306	104421			19\$: TCHKOP		;CHECK NO ERRORS SET
3168	004310	104004			ERROR	4 ;OR5,6,7	;REPORT ALL ERRORS
3169							
3170	004312	032737	100000	001552	BIT	#SVAL,T.DS	;CHECK SVAL SET
3171	004320	001004			BNE	20\$	;YES - SKIP
3172	004322	012737	055506	001400	MOV	#EM48,EM3N	;MESSAGE (NO SVAL FROM EXISTANT DR)
3173	004330	104003			ERROR	3	
3174							
3175	004332	012737	000103	001600	20\$: MOV	#PACK,L.CS1	;ELSE SET TO DO PACK ACK

M05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 63
T3 SELECT ALL DRIVES

SEQ 0063

```

3176 004340 104417          TLOADRK          ;LOAD RK
3177                                ;WAIT FOR INTERRUPT
3178 004342 104423          TWAT16          ;TO SLOW/NOT COMPLETE ERROR
3179 004344 104002          ERROR 2
3180                                ;CHECK FOR ANY ERRORS
3181 004346 104421          TCHKOP          ;YES - REPORT & ABORT TEST
3182 004350 104004          ERROR 4 ;OR5,6,7
3183                                ;DID VV SET
3184 004352 032737 000100 001552          BIT  #VV,T.DS          ;YES - SKIP
3185 004360 001005          BNE 22$
3186 004362 012737 055370 001400          MOV  #EM46,EM3N          ;MESSAGE (VV DID NOT SET/
3187 004370 104003          ERROR 3
3188 004372 000734          BR 18$
3189
3190 004374 032737 040000 001552 22$:      BIT  #DSC,T.DS          ;TEST IF DSC RESET
3191 004402 001410          BEQ 23$          ;YES - SKIP
3192 004404 012737 056502 001470          MOV  #EMDSC,EM12N
3193 004412 012737 056742 061010          MOV  #EMSCLR,DF011A
3194 004420 104003          ERROR 3          ;"DSC NOT RESET RESULT OF SUBSYS CLEAR"
3195 004422 000720          BR 18$
3196
3197 004424 005737 001630          23$:      TST  DRVBIT          ;TEST IF DRVBIT IS NEGATIVE
3198 004430 100315          BPL 18$          ;NO - SKIP
3199 004432 010137 001630          MOV  R1,DRVBIT          ;STORE DRIVE SELECT BIT
3200 004436 010037 001240          MOV  R0,$TMP7          ;STORE DRIVE NUMBER TO BE TESTED
3201 004442 000710          BR 19$
3202
3203 004444 013737 001240 001626 21$:      MOV  $TMP7,DRVNUM          ;LOAD LOWEST # DRIVE PRESENT INTO DRVNUM
3204
3205 004452 023727 001624 000002          CMP  SRTFLG,#2          ;TEST IF CLOCK ADJUST START
3206 004460 001002          BNE 25$          ;NO - SKIP
3207 004462 000137 043274          JMP  ADJCLK          ;GO TO ADJUST CLOCK ROUTINE
3208
3209 004466          25$:
3210
3211          ;*****
3212          ;TEST 4          RELEASE ALL DRIVES
3213          ;
3214          ;          RELEASE ALL DRIVES. MAKE SURE NO ERROR
3215          ;          SETS AND STATUS VALID IS RESET.
3216          ;
3217          ;*****
3218          ;*****
3219          ;*****
3220          ;*****
3221          ;*****
3222          ;*****
3223          ;*****
3224          ;*****
3225          ;*****
3226          ;*****
3227          ;*****
3228          ;*****
3229          ;*****
3230          ;*****
3231          ;*****

```

N05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 64
T4 RELEASE ALL DRIVES

SEQ 0064

```

3232 004530 012737 000010 001610      MOV      #RLS,L.CS2      ;SET DRIVE RELEASE,STILL SET FOR SELECT
3233
3234 004536 104417      TLOADRK      ;LOAD RK REGS
3235 004540 104423      TWAT16      ;WAIT FOR INTERRUPT
3236 004542 104002      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3237
3238
3239 004544 104421      TCHKOP      ;CHECK FOR ANY ERRORS
3240 004546 104004      ERROR      4 ;OR 5, 6, OR 7 ;REPORT ALL ERRORS
3241 004550 032737 100000 001552      BIT      #SVAL,T.DS      ;DID SVAL RESET?
3242 004556 001404      BEQ      1$      ;YES - SKIP
3243 004560 012737 055565 001400      MOV      #EM49,EM3N      ;MESSAGE (SVAL NOT RESET W/RELEASE)
3244 004566 104003      ERROR      3
3245
3246
3247 004570      1$:
3248
3249      TSTLUP:
3250      *****
3251      *TEST 5      NON-STANDARD MESSAGES AND SVAL
3252      *
3253      *      PICK ONE OF THE AVAILABLE DRIVES AND GET
3254      *      NON-STANDARD MESSAGES. MAKE SURE NO
3255      *      ERROR OCCURS AND STATUS VALID DOES NOT SET
3256      *      AND THAT NON-STANDARD MESSAGES CAUSE STATUS
3257      *      VALID TO RESET.
3258      *****
3259      *ST5:      SCOPE
3260      MOV      #50.,$TIMES      ;DO 50. ITERATIONS
3261      TSSINIT      ;CLEAR SUBSYSTEM
3262      ERROR      3      ;BAD CLEAR MESSAGE
3263      MOV      #1,R1      ;PRESET R1 FOR MESSAGE PAIR 1
3264      MOV      #DRVNUM,L.CS2      ;LOAD DRV NUMBER
3265      MOV      #SELD,DRV,L.CS1      ;LOAD SELECT COMMAND
3266      CLR      L.MR1      ;LOAD FOR STANDARD STATUS
3267      TLOADRK      ;LOAD RK
3268      TWAT16      ;WAIT FOR INTERRUPT
3269      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3270      TCHKOP      ;CHECK OPERATION
3271      ERROR      4 ;5,6 OR 7      ;REPORT ALL ERRORS
3272
3273 004642 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST STATUS VALID SET
3274 004650 001007      BNE      2$      ;YES-SKIP
3275
3276 004652 012737 056342 001460      MOV      #EMSVAL,EM11N
3277 004660 012737 047640 061010      MOV      #EMSELD,DF011A
3278 004666 104011      ERROR      11      ;"SVAL NOT SET RESULT OF DRIVE SELECT"
3279
3280 004670 010137 001616      2$:      MOV      R1,L.MR1      ;LOAD MESSAGE PAIR SELECT
3281
3282      TLOADRK      ;LOAD RK
3283      TWAT16      ;WAIT FOR INTERRUPT
3284      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3285
3286      TCHKOP      ;CHECK OPERATION
3287      ERROR      4 ;5,6, OR 7      ;REPORT ALL ERRORS

```

B06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 65
TS NON-STANDARD MESSAGES AND SVAL

SEQ 0065

```

3298 004706 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST STATUS VALID RESET
3299 004714 001407                      BEQ      3$          ;YES-SKIP
3290
3291 004716 012737 056342 001470      MOV      #EMSVAL,EM12N
3292 004724 012737 056357 061010      MOV      #EMNZPR,DF011A
3293 004732 104012                      ERROR    12          ;"SVAL NOT RESET RESULT OF SEL W/ NON-0 PAIR"
3294
3295 004734 022701 000003      3$:    CMP      #3,R1          ;WAS PAIR 3 SELECTED?
3296 004740 001402                      BEQ      4$          ;YES-SKIP
3297 004742 005201                      INC      R1          ;BUMP TO NEXT PAIR
3298 004744 000727                      BR       1$          ;SKIP TO DO IT.
3299 004746
3300
3301      ;*****
3302      ;TEST 6      WRITING CS2 AND STATUS VALID
3303      ;
3304      ;      SELECT AN AVAILABLE DRIVE.  MAKE SURE STATUS
3305      ;      VALID IS SET.  WRITE COMMAND AND STATUS REGISTER 2.
3306      ;      MAKE SURE STATUS VALID RESETS.
3307      ;*****
3308 004746 000004      1$T6:    SCOPE
3309 004750 012737 000062 001262      MOV      #50.,$TIMES      ;DO 50. ITERATIONS
3310 004756 104416                      TSSINIT
3311 004760 104003                      ERROR    3          ;CLEAR SUBSYSTEM
3312      ;                      ;BAD INIT ERROR
3313 004762 013737 001626 001610      MOV      DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3314 004770 012737 000101 001600      MOV      #SELDV,L.CS1      ;LOAD DRIVE SELECT
3315
3316 004776 104417                      TLOADRK      ;LOAD RK
3317 005000 104423                      TWAIT6      ;WAIT FOR INTERRUPT
3318 005002 104002                      ERROR    2          ;TO SLOW/NOT COMPLETE ERROR
3319
3320 005004 104421                      TCHKOP
3321 005006 104004                      ERROR    4 ;5,6, OR 7 ;CHECK OPERATION
3322      ;                      ;REPORT ALL ERRORS
3323 005010 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST STATUS VALID SET
3324 005016 001007                      BNE      1$          ;YES-SKIP
3325
3326 005020 012737 056342 001460      MOV      #EMSVAL,EM11N
3327 005026 012737 047640 061010      MOV      #EMSELD,DF011A
3328 005034 104011                      ERROR    11          ;"SVAL NOT SET RESULT OF DRV SELECT"
3329
3330 005036 013762 001626 000010      1$:    MOV      DRVNUM,RKCS2(R2) ;WRITE CS2 TO RESET SVAL
3331
3332 005044 104420                      TGETRK          ;GET RK REGS.
3333
3334 005046 032737 100000 001552      BIT      #SVAL,T.DS      ;TEST SVAL RESET
3335 005054 001407                      BEQ      2$          ;YES-SKIP
3336
3337 005056 012737 056342 001470      MOV      #EMSVAL,EM12N
3338 005064 012737 056420 061010      MOV      #EMWCS2,DF011A
3339 005072 104012                      ERROR    12          ;"SVAL NOT RESET BY WRITING CS2"
3340
3341      2$:

```

C06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 66
**CONTROLLER ERROR TESTS

SEQ 0066

.SBTTL **CONTROLLER ERROR TESTS

*TEST 7 DRIVE TYPE ERROR

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

*ST7: SCOPE

MOV #50., \$TIMES ; DO 50. ITERATIONS
TSSINIT ; CLEAR SUBSYSTEM
ERROR 3 ; BAD INIT ERROR

MOV DRVNUM, L.CS2 ; LOAD DRIVE NUMBER
MOV #SELDIV, L.CS1 ; LOAD DRIVE SELECT
BIS #CDT, L.CS1 ; LOAD DRIVE TYPE

TLOADRK ; LOAD RK
TWAIT6 ; 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 IS ; YES-SKIP

MOV #EMSVAL, EM11N
MOV #EMDTP, DFO11A
ERROR 11 ; "SVAL NOT SET RESULT OF DRV TYPE ERR"

IS: *****
*TEST 10 STATUS VALID AND PARITY ERROR

*
* ISSUE A SELECT TO AN AVAILABLE 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 AVAILABLE DRIVES.
*

*ST10: SCOPE

MOV #50., \$TIMES ; DO 50. ITERATIONS
TSSINIT ; CLEAR SUBSYSTEM

3342
3343
3344
3345
3346
3347
3348
3349
3350
3351 005074 000004
3352 005076 012737 000062 001262
3353 015104 104416
3354 005106 104003
3355
3356 005110 013737 001626 001610
3357 005116 012737 000101 001600
3358 005124 052737 002000 001600
3359
3360 005132 104417
3361 005134 104423
3362 005136 104002
3363
3364 005140 104422
3365 005142 000040
3366 005144 000000
3367 005146 000000
3368 005150 104004
3369
3370 005152 032737 100000 001552
3371 005160 001007
3372
3373 005162 012737 056342 001460
3374 005170 012737 055036 061010
3375 005176 104011
3376 005200
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395 005200 000004
3396 005202 012737 000062 001262
3397 005210 104416

D06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 67
T10 STATUS VALID AND PARITY ERROR

SEQ 0067

3398	005212	104003			ERROR 3		;BAD INIT ERROR
3399							
3400	005214	013737	001626	001610	MOV	DRVNUM,L.CS2	;LOAD DRIVE NUMBER
3401	005222	012737	000101	001600	MOV	#SELDIV,L.CS1	;LOAD DRIVE SELECT
3402	005230	012737	000020	001616	MOV	#PAT,L.MR1	;LOAD EVEN PARITY BIT
3403							
3404	005236	104417			TLOADRK		;LOAD RK REGS-SELECT W/EVEN PARITY
3405	005240	104423			TWAIT6		;WAIT FOR INTERRUPT
3406	005242	104002			ERROR 2		;TO SLOW/NOT COMPLETE ERROR
3407							
3408	005244	104422			TCHKWE		;CHECK OPERATION FOR EXPECTED ERROR
3409	005246	000011			DRPARERR:SPARERR		;DRIVE SELECTED DRIVE BUS PARITY ERROR
3410	005250	000000			.WORD 0		;CONTROLLER DETECTED DRIVE BUS PARITY ERROR
3411	005252	000000			.WORD 0		
3412	005254	104004			ERROR 4 ; OR 5,6,7		;REPORT ANY DIFFERENCES
3413							
3414	005256	012700	000400		MOV	#BIT8,RO	;ROUTINE TO DETERMINE WHICH BIT
3415							
3416	005262	013701	001626		MOV	DRVNUM,R1	;SHOULD BE SET IN ASOF TO INDICATE
3417	005266	001403			BEQ 3\$		;DRIVE ATTENTION. RO WILL HAVE THE
3418	005270	006300			ASL RO		;BIT THAT SHOULD BE SET FOR THE DRIVE
3419	005272	005301			DEC R1		;IN USE
3420	005274	001375			BNE 2\$		
3421							
3422	005276	030037	001556		3\$: BIT RO,T.ASOF		;TEST ATTENTION SET
3423	005302	001007			BNE 4\$		;YES-SKIP
3424	005304	012737	056526	001460	MOV	#EMDA,EM11N	
3425	005312	012737	054753	061010	MOV	#EMDPA,DF011A	
3426	005320	104011			ERROR 11		; "DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3427	005322	032737	040000	001540	4\$: BIT #DI,T.CS1		;TEST DRIVE INTERRUPT SET
3428	005330	001007			BNE 5\$		;YES-SKIP
3429	005332	012737	056462	001460	MOV	#EMDI,EM11N	
3430	005340	012737	054753	061010	MOV	#EMDPA,DF011A	
3431	005346	104011			ERROR 11		; "DRV INT NOT SET RESULT OF DRV PARITY ERR"
3432							
3433	005350	032737	040000	001552	5\$: BIT #DSC,T.DS		;TEST DRIVE STATUS CHANGE SET
3434	005356	001007			BNE 6\$		;YES-SKIP
3435	005360	012737	056502	001460	MOV	#EMOSC,EM11N	
3436	005366	012737	054753	061010	MOV	#EMDPA,DF011A	
3437	005374	104011			ERROR 11		; "DSC NOT SET RESULT OF DRV PARITY ERR"
3438							
3439	005376	032737	100000	001552	6\$: BIT #SVAL,T.DS		;TEST STATUS VALID SET
3440	005404	001007			BNE 7\$		;YES-SKIP
3441	005406	012737	056342	001460	MOV	#EMSVAL,EM11N	
3442	005414	012737	054753	061010	MOV	#EMDPA,DF011A	
3443							
3444	005422	104011			ERROR 11		; "SVAL NOT SET RESULT OF DRV PAR ERR"
3445							
3446	005424	005037	001616		7\$: CLR L.MR1		;CLEAR PAT IN MR1
3447							
3448	005430	052737	100000	001600	BIS #CCLR,L.CS1		;CLEAR CONTROLLER
3449	005436	104417			TLOADRK		;LOAD RK REGS TO DO CLEAR
3450							
3451							
3452	005440	104421			TCHKOP		;CHECK NO ERRORS SET
3453	005442	104004			ERROR 4 ; OR 5,6,7		;REPORT ALL ERRORS STILL SET

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRSKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 68
T10 STATUS VALID AND PARITY ERROR

3454	005444	030037	001556		BIT	RO,T.ASOF	;TEST ATTENTION STILL SET
3455	005450	001007			BNE	8\$	;YES-SKIP
3456	005452	012737	056526	001510	MOV	#EMDA,EM14N	
3457	005460	012737	056546	061010	MOV	#EMCCLR,DF011A	
3458	005466	104014			ERROR	14	; "DRV ATT RESET RESULT OF CONT CLR"
3459							
3460	005470	032737	040000	001540	8\$: BIT	#DI,T.CS1	;TEST DRIVE INTERRUPT STILL SET
3461	005476	001007			BNE	9\$	;YES-SKIP
3462	005500	012737	056462	001510	MOV	#EMDI,EM14N	
3463	005506	012737	056546	061010	MOV	#EMCCLR,DF011A	
3464	005514	104014			ERROR	14	; "DRV INT RESET RESULT OF CONTROLLER CLR"
3465							
3466	005516	012737	000101	001600	9\$: MOV	#SELDIV,L.CS1	;LOAD SELECT DRIVE
3467							
3468	005524	104417			TLOADRK		;LOAD RK REGS
3469	005526	104423			TWAIT6		;WAIT FOR INTERRUPT
3470	005530	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
3471							
3472	005532	104422			TCHKWE		;CHECK THAT ERRORS ARE SET
3473	005534	000010			.WORD	000010	;DPAI SET (NO SPAR 1, REPORT
3474	005536	000000			.WORD	0	;ANY DISCREPENCIES
3475	005540	000000			.WORD	0	
3476	005542	104004			ERROR	4 ; OR 5,6,7	
3477	005544	030037	001556		BIT	RO,T.ASOF	;TEST ATTENTION STILL SET.
3478	005550	001007			BNE	10\$	;YES-SKIP
3479	005552	012737	056526	001460	MOV	#EMDA,EM11N	
3480	005556	012737	054753	061010	MOV	#EMDPAI,DF011A	
3481	005566	104011			ERROR	11	; "DRV ATT NOT SET RESULT OF DRV PARITY ERR"
3482							
3483	005570	032737	040000	001540	10\$: BIT	#DI,T.CS1	;TEST DRIVE INTERRUPT STILL SET
3484	005576	001007			BNE	11\$	;YES-SKIP
3485	005600	012737	056462	001460	MOV	#EMDI,EM11N	
3486	005606	012737	054753	061010	MOV	#EMDPAI,DF011A	
3487	005614	104011			ERROR	11	; "DRV INT NOT SET RESULT OF DRV PARITY ERR"
3488							
3489	005616	032737	040000	001552	11\$: BIT	#DSC,T.DS	;TEST STATUS CHANGE STILL SET
3490	005624	001007			BNE	12\$	;YES-SKIP
3491	005626	012737	056502	001460	MOV	#EMDSC,EM11N	
3492	005634	012737	054753	061010	MOV	#EMDPAI,DF011A	
3493	005642	104011			ERROR	11	; "DSC NOT SET RESULT OF DRV PARITY ERR"
3494							
3495	005644	032737	100000	001552	12\$: BIT	#SVAL,T.DS	;TEST STATUS VALID STILL SET
3496	005652	001007			BNE	13\$	;YES-SKIP
3497	005654	012737	056342	001460	MOV	#EMSVAL,EM11N	
3498	005662	012737	054753	061010	MOV	#EMDPAI,DF011A	
3499	005670	104011			ERROR	11	; "SVAL NOT SET RESULT OF DRV PARITY ERR"
3500							
3501	005672	052737	100000	001600	13\$: BIS	#CCLR,L.CS1	;CLEAR CONTROLLER
3502							
3503	005700	104417			TLOADRK		;LOAD RK REGS (DO CLEAR)
3504							
3505	005702	012701	000001		MOV	#1,R1	;SET COUNTER TO SELECT STATUS PAIR
3506	005706	012737	000101	001600	MOV	#SELDIV,L.CS1	;LOAD DRIVE SELECT
3507							
3508							
3509							

F06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 69
T10 STATUS VALID AND PARITY ERROR

SEQ 0069

```

3510 005714 010137 001616      14$: MOV R1,L.MR1      ;LOAD STATUS PAIR SELECTION
3511 005720 104417              TLOADRK      ;LOAD RK REGS
3512 005722 104423              TWAT16      ;WAIT FOR INTERRUPT
3513 005724 104002              ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
3514
3515 005726 104421              TCHKOP      ;CHECK IF ANY ERRORS SET
3516 005730 104004              ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS SET.
3517
3518 005732 030037 001556      BIT RO,T.ASOF      ;TEST ATTENTION STILL SET
3519 005736 001007              BNE 15$      ;YES-SKIP
3520 005740 012737 056526 001410  MOV #EMDA,EM4N
3521 005746 012737 056357 061010  MOV #EMNZPR,DF011A
3522 005754 104014              ERROR 14      ;"ATTENTION RESET RESULT OF NON-0 PAIR SEL"
3523
3524 005756 032737 040000 001540 15$: BIT #DI,T.CS1
3525 005764 001007              BNE 16$
3526 005766 012737 056462 001510  MOV #EMDI,EM14N
3527 005774 012737 056357 061010  MOV #EMNZPR,DF011A
3528 006002 104014              ERROR 14      ;"DRV INT RESET RESULT OF NON-0 PAIR SELECT"
3529
3530 006004 005201              16$: INC R1      ;BUMP PAIR SELECT
3531 006006 022701 000004      CMP #4,R1      ;ALL PAIRS DONE?
3532 006012 001340              BNE 14$      ;NO-LOOP
3533
3534 006014 005037 001616      CLR L.MR1      ;CLEAR MR1
3535
3536 006020 012737 000105 001600  MOV #CLEAR,L.CS1 ;LOAD DRIVE CLEAR
3537
3538 006026 104417              TLOADRK      ;DO DRIVE CLEAR
3539 006030 104423              TWAT16      ;WAIT FOR INTERRUPT
3540 006032 104002              ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
3541
3542 006034 104421              TCHKOP      ;CHECK FOR ANY ERRORS
3543 006036 104004              ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
3544
3545 006040 012701 000020      17$: MOV #20,R1      ;SET COUNT FOR SHORT WAIT
3546 006044 005301              DEC R1      ;TO ALLOW CONTROLLER TIME TO POLL
3547 006046 001376              BNE 17$      ;DRIVES
3548
3549 006050 104420              TGETRK      ;GET RK REGS
3550 006052 030037 001556      BIT RO,T.ASOF      ;TEST ATTENTION RESET
3551 006056 001407              BEQ 18$      ;YES-SKIP
3552 006060 012737 056526 001470  MOV #EMDA,EM12N
3553 006066 012737 047666 061010  MOV #EMDCLR,DF011A
3554 006074 104012              ERROR 12      ;"ATTENTION NOT RESET RESULT OF DRV CLEAR
3555
3556 006076 032737 040000 001540 18$: BIT #DI,T.CS1      ;TEST DRIVE INTERRUPT RESET
3557 006104 001407              BEQ 19$      ;YES-SKIP
3558 006106 012737 056462 001470  MOV #EMDI,EM12N
3559 006114 012737 047666 061010  MOV #EMDCLR,DF011A
3560 006122 104012              ERROR 12      ;"DRV INT NOT RESET RESULT OF DRIVE CLR"
3561
3562 006124              19$:
3563 ;*****
3564 ;*TEST 11 UNIT FIELD ERROR ON RELEASE
3565 ;*
```

G06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 70
T11 UNIT FIELD ERROR ON RELEASE

SEQ 0070

```

3566      *      ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
3567      *      DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
3568      *      A RELEASE COMMAND. CLOCK THROUGH PHASE ADDRESS 2.
3569      *      TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD
3570      *      ERROR SETS.
3571      *
3572      * *****
3573      * ST11: SCOPE
3574      *      MOV      #50.,$TIMES      ;DO 50. ITERATIONS
3575      *      TSSINIT      ;CLEAR SUBSYSTEM
3576      *      ERROR      2      ;BAD INIT ERROR
3577
3578      *      MOV      DRVNUM,L.CS2      ;SELECT A DRIVE
3579      *      MOV      #SELDV,L.CS1      ;DO DRIVE SELECT
3580
3581      *      TLOADRK      ;LOAD RK
3582      *      TWAIT6      ;WAIT FOR INTERRUPT
3583      *      ERROR      2      ;TO SLOW/NOT COMPLETE ERROR
3584
3585      *      TCHKOP      ;CHECK FOR ANY ERRORS
3586      *      ERROR      4 ; OR 5,6,7 ;REPORT ALL ERRORS.
3587
3588      *      BIS      #RLS,L.CS2      ;LOAD RELEASE
3589      *      MOV      #DMD,L.MR1      ;SET DIAGNOSTIC MODE
3590
3591      *      TLOADRK      ;LOAD RK
3592
3593      *      JSR      R4,MCLOCK      ;CALL MAINT CLOCK
3594      *      .BYTE      #D19      ;NUMBER OF PHASES
3595      *      .BYTE      2      ;NUMBER OF CLOCK XISTIONS
3596
3597      *      BIC      #DMD,RKMR1(R2) ;CLEAR DIAG MODE
3598
3599      *      TWAIT6      ;WAIT FOR INTERRUPT
3600      *      ERROR      2      ;TO SLOW/NOT COMPLETED
3601
3602      *      TCHKWE      ;CHECK OPERATION WITH ERROR
3603      *      .WORD      0
3604      *      .WORD      0
3605      *      .WORD      UFERK      ;UNIT FIELD ERROR
3606      *      ERROR      4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
3607
3608      *      TSSINIT      ;CLEAR SUBSYSTEM TO INSURE UFE RESETS
3609      *      ERROR      2
3610      * *****
3611      * TEST 12      UNIT FIELD ERROR ON SELECT
3612      *
3613      *      ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
3614      *      DRIVE. PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE
3615      *      A SELECT COMMAND WITH MESSAGE ID = 3 AND DRIVE
3616      *      SELECTED = 0. CLOCK THROUGH PHASE ADDRESS 6.
3617      *      TURN OFF DIAGNOSTIC MODE. MAKE SURE UNIT FIELD
3618      *      ERROR SETS.
3619      *
3620      * *****
3621      * ST12: SCOPE

```

H06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 71
T12 UNIT FIELD ERROR ON SELECT

SEQ 0071

3622	006244	012737	000062	001262	MOV	#50.,\$TIMES	::DO 50. ITERATIONS
3623	006252	104416			TSSINIT		::CLEAR SUBSYSTEM
3624	006254	104003			ERROR	3	::BAD INIT ERROR
3625							
3626	006256	013737	001626	001610	MOV	DRVNUM,L.CS2	::LOAD DRIVE NUMBER
3627	006264	012737	000101	001600	MOV	#SELDV,L.CS1	::LOAD DRIVE SELECT
3628							
3629	006272	104417			TLOADRK		::LOAD RK
3630	006274	104423			TWAT16		::WAIT FOR INTERRUPT
3631	006276	104002			ERROR	2	::TO SLOW/NOT COMPLETE
3632							
3633	006300	104421			TCHKOP		::CHECK FOR ANY ERROR
3634	006302	104004			ERROR	4 ; OR 5,6,7	::REPORT ALL ERRORS
3635							
3636	006304	012737	000043	001616	MOV	#DMD!BIT1!BIT0,L.MR1	::LOAD DIAG MODE & MSG PAIR 3
3637	006312	005037	001610		CLR	L.CS2	::LOAD FOR DRIVE 0
3638							
3639	006316	104417			TLOADRK		::LOAD RK
3640							
3641	006320	004437	036004		JSR	R4,MCLOCK	::CALL MAINTENANCE CLOCK
3642	006324	026			.BYTE	1022	::THROUGH PHASE 6
3643	006325	002			.BYTE	2	::PLUS 2 TRANSITIONS
3644							
3645	006326	042762	000040	000026	BIC	#DMD,RKMR1(R2)	::CLEAR DIAG MODE
3646							
3647	006334	104423			TWAT16		::WAIT FOR INTERRUPT
3648	006336	104002			ERROR	2	::TO SLOW/NOT COMPLETED ERROR
3649							
3650	006340	104422			TCHKWE		::CHECK OPERATION WITH ERROR
3651	006342	000000			.WORD	0	
3652	006344	000000			.WORD	0	
3653	006346	000004			.WORD	UFERR	::UNIT FIELD ERROR SHOULD SET
3654	006350	104004			ERROR	4 ; OR 5,6,7	::REPORT ANY DISCREPENCIES
3655							
3656							
3657							
3658							
3659							
3660							
3661							
3662							
3663							
3664							
3665							
3666							
3667							
3668							
3669							
3670							
3671	006352	000004			ST13:	SCOPE	
3672	006354	012737	000062	001262	MOV	#50.,\$TIMES	::DO 50. ITERATIONS
3673	006362	104416			TSSINIT		::CLEAR SUBSYSTEM
3674	006364	104003			ERROR	3	::BAD INIT ERROR
3675							
3676	006366	013737	001626	001610	MOV	DRVNUM,L.CS2	::LOAD DRIVE NUMBER
3677	006374	012737	000113	001600	MOV	#RECAL,L.CS1	::LOAD RECAL

.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)
*****

```

```

3678
3679 006402 104417 TLOADRK ;LOAD RK
3680 006404 104423 TWAT16 ;WAIT FOR 1ST INTERRUPT
3681 006406 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
3682 006410 005037 001662 CLR INTSET ;CLEAR INTERRUPT FLAG
3683 006414 104420 TGETRK ;GET RK REGS
3684 006416 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAL SET
3685 006424 001010 BNE 1$ ;YES-SKIP
3686 006426 012737 056342 001460 MOV #EMSVAL,EM11N
3687 006434 012737 047727 061010 MOV #EMRCAL,DF011A
3688 006442 104011 ERROR 11 ;"SVAL NOT SET RESULT OF RECAL"
3689 006444 000463 BR 50$ ;ABORT TEST
3690
3691 006446 104437 1$: TWAT8S ;WAIT FOR INTERRUPT
3692 006450 000401 BR 2$ ;NO INTERRUPT RETURN
3693 006452 000404 BR 3$ ;INTERRUPT RETURN
3694
3695 006454 012737 055635 001370 2$: MOV #EM50,EM2N ;ALTER MESSAGE "NO 2ND INTERRUPT OR IT WAS LATE"
3696 006462 104002 ERROR 2
3697
3698 006464 104420 3$: TGETRK ;GET RK REGS
3699 006466 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAC SET NOW
3700 006474 001410 BEQ 4$ ;NO-SKIP
3701 006476 012737 056342 001470 MOV #EMSVAL,EM12N
3702 006504 012737 056567 061010 MOV #EM2INT,DF011A
3703 006512 104012 ERROR 12 ;"SVAL NOT RESET RESULT OF SECOND TEST"
3704 006514 000437 BR 50$
3705
3706 006516 032737 040000 001552 4$: BIT #DSC,T.DS ;TEST DSC SET BY ATTENTION
3707 006524 001010 BNE 5$ ;YES-SKIP
3708 006526 012737 056502 001460 MOV #EMDSC,EM11N
3709 006534 012737 056567 061010 MOV #EM2INT,DF011A
3710 006542 104011 ERROR 11 ;"DSC NOT SET RESULT OF SECOND INTERRUPT"
3711 006544 000423 BR 50$
3712
3713 006546 012737 000101 001600 5$: MOV #SELDIV,L.CS1 ;LOAD DRIVE SELECT
3714
3715 006554 104417 TLOADRK ;LOAD RK REGS
3716 006556 104423 TWAT16 ;WAIT FOR INTERRUPT
3717 006560 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
3718
3719 006562 104421 TCHKOP ;CHECK FOR ANY ERRORS
3720
3721 006564 104004 ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
3722
3723 006566 032737 100000 001552 BIT #SVAL,T.DS ;TEST SVAC SET
3724 006574 001007 BNE 50$ ;YES-SKIP
3725 006576 012737 056342 001460 MOV #EMSVAL,EM11N
3726 006604 012737 047640 061010 MOV #EMSELD,DF011A
3727 006612 104011 ERROR 11 ;"SVAL NOT SET RESULT OF DRV SEL.
3728 006614
3729
3730 *****
3731 ;TEST 14 SINGLE INTERRUPT FROM ATTENTION
3732 ;
3733 ; DO A SEEK TO CYLINDER 0. WAIT FOR INTERRUPT FROM
; DRIVE ATTENTION. LOWER PRIORITY AGAIN AND MAKE

```

J06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZ6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 73
T14 SINGLE INTERRUPT FROM ATTENTION

SEQ 0073

```

3734
3735
3736
3737 006614 000004
3738 006616 012737 000062 001262
3739 006624 104416
3740 006626 104003
3741
3742 006630 013737 001626 001610
3743 006636 012737 000117 001600
3744
3745 006644 104417
3746 006646 104423
3747 006650 104002
3748
3749
3750 006652 104420
3751
3752 006654 032737 040000 001540
3753 006662 001010
3754 006664 012737 056462 001460
3755 006672 012737 056610 061010
3756 006700 104011
3757 006702 000417
3758
3759
3760 006704 012700 000031
3761 006710 005300
3762 006712 001376
3763
3764 006714 022737 000001 001662
3765 006722 001407
3766 006724 012737 056762 001500
3767 006732 012737 056610 051010
3768 006740 104013
3769
3770 006742 104421
3771 006744 104004
3772
3773
3774
3775
3776
3777
3778
3779 006746 000004
3780 006750 012737 000005 001262
3781 006756 104416
3782 006760 104003
3783
3784 006762 013737 001626 001610
3785 006770 012737 000117 001600
3786
3787 006776 104417
3788 007000 104423
3789 007002 104002

; * SURE ANOTHER INTERRUPT DOES NOT OCCUR. CLEAR DRIVE.
; *
; *****
; ST14: SCOPE
; MOV #50., $TIMES ; DO 50. ITERATIONS
; TSSINIT ; CLEAR SUBSYSTEM
; ERROR 3 ; BAD INIT ERROR

; MOV DRVNUM, L.CS2 ; LOAD DRIVE NUMBER
; MOV #SEEK, L.CS1 ; LOAD SEEK DCYL LEFT AT 0.

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

; TGETRK ; GET RK REGS

; BIT #DI, T.CS1 ; TEST DI SET
; BNE 2$ ; YES-SKIP
; MOV #EMDI, EM11N
; MOV #EMSKSF, DF011A
; ERROR 11 ; "DI NOT SET RESULT OF SEEK TO SELF"
; BR 50$

2$: MOV #25., R0 ; LOAD AND DECREMENT A COUNT TO
3$: DEC R0 ; ZERO. GIVE CONTROLLER A CHANCE TO
; BNE 3$ ; INTERRUPT AGAIN. ERROR IF IT DOES.

; CMP #1, INTSET ; CHECK ONLY ONE INTERRUPT OCCURRED
; BEQ 50$ ; YES-SKIP
; MOV #EMMI, EM13N
; MOV #EMSKSF, DF011A
; ERROR 13 ; "MULTIPLE INTERRUPTS RESULT OF SEEK TO SELF"

50$: TCHKOP ; CHECK FOR ANY ERRORS
; ERROR 4 ; OR 5,6,7 ; REPORT ALL ERRORS

; *****
; 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.
; *
; *****
; ST15: SCOPE
; MOV #5., $TIMES ; DO 5. ITERATIONS
; TSSINIT ; CLEAR SUBSYSTEM
; ERROR 3 ; BAD INIT ERROR

; MOV DRVNUM, L.CS2 ; LOAD DRIVE NUMBER
; MOV #SEEK, L.CS1 ; LOAD SEEK (TO SELF-0)

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

```

K06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 74
T15 RESET ATTENTIONS WITH UNIBUS INIT

SEQ 0074

```

3790
3791 007004 104420          TGETRK          ;GET RK REGS
3792
3793 007006 032737 040000 001540      BIT      #DI,T.CS1      ;TEST DI SET
3794 007014 001010          BNE      1$          ;YES-EXIT
3795 007016 012737 056462 001460      MOV      #EMDI,EM11N
3796 007024 012737 056610 061010      MOV      #EMSKSF,DF011A
3797 007032 104011          ERROR      11          ;"DI NOT SET RESULT OF SEEK TO SELF
3798 007034 000450          BR      50$
3799
3800 007036 005037 001662          1$: CLR      INTSET          ;CLEAR INTERRUPT COUNTER
3801 007042 000005          RESET          ;DO UNIBUS RESET
3802 007044 004737 044626          JSR      PC,$TKINT        ;RESET KEYBOARD INTERRUPT
3803
3804 007050 005037 001674          CLR      LCLKTK          ;CLEAR TICK COUNTER
3805 007054 004737 035150          JSR      PC,MYTIME        ;CALL TIMER
3806 007060 022737 000012 001674      CMP      #10.,LCLKTK      ;COUNT 10 TICKS (MILLISECONDS)?
3807 007066 001372          BNE      5$          ;NO - LOOP
3808
3809 007070 012762 000100 000000      MOV      #IE,RKCS1(R2)    ;SET IE FOR ANY STRAY INTERRUPTS
3810 007076 004737 034434          JSR      PC,$PTTST        ;SET UP OPTIONS AGAIN
3811
3812 007102 104423          TWAT16          ;WAIT 16 MS FOR AN INTERRUPT
3813 007104 000410          BR      2$          ;NONE IS EXPECTED SO RETURN SHOULD BE
3814                                     ;HERE-BR TO CONTINUE TEST.
3815 007106 012737 056625 001500      MOV      #EMUXIT,EM13N    ;INT OCCURRED ON RESET
3816 007114 012737 056667 061010      MOV      #EMRSET,DF011A
3817 007122 104013          ERROR      13          ;"UNEXECUTED INTERRUPT RESULT OF RESET"
3818 007124 000414          BR      50$
3819 007126 104420          2$: TGETRK          ;GET RK REGS
3820 007130 032737 040000 001540      BIT      #DI,T.CS1      ;TEST DI RESET
3821 007136 001407          BEQ      50$          ;YES-SKIP
3822 007140 012737 056462 001470      MOV      #EMDI,EM12N
3823 007146 012737 056667 061010      MOV      #EMRSET,DF011A
3824 007154 104012          ERROR      12          ;"DI NOT RESET RESULT OF RESET"
3825
3826 007156          50$:
3827
3828 .SBTTL **ILLEGAL DISK ADDRESS ERROR TESTS
3829
3830 ;*****
3831 ;TEST 16      ILLEGAL DISK ADDRESS (PART 1)
3832 ;
3833 ;      ISSUE A SEEK TO CYLINDER 0, HEAD 3. MAKE SURE
3834 ;      ILLEGAL ADDRESS ERROR AND SEEK INCOMPLETE SETS.
3835 ;      CLEAR CONTROLLER AND CLEAR DRIVE. REPEAT FOR HEADS 4-7,
3836 ;      CHECKING THAT BOTH IDAE AND SEEK INCOMPLETE SET FOR
3837 ;      HEAD 7 AND IDAE ALONE SETS FOR HEADS 4, 5, AND 6.
3838 ;*****
3839
3840 007156 000004          TST16: SCOPE
3841 007160 012737 000012 001262      MOV      #10,$TIMES      ;DO 10. ITERATIONS
3842 007166 012701 000003          MOV      #3,R1          ;PRESET FOR SELECTING TRACK 3
3843
3844 007172 104416          TSSINIT          ;CLEAR SUBSYSTEM
3845 007174 104003          ERROR      3

```

L06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 75
T16 ILLEGAL DISK ADDRESS (PART 1)

SEQ 0075

```

3846
3847 007176 012737 000113 001600      MOV    #RECAL,L.CS1      ;SET UP TO RECAL
3848 007204 013737 001626 001610      MOV    DRVNUM,L.CS2      ;LOAD DRIVE
3849
3850 007212 104417                      TLOADRK      ;LOAD RK REGS
3851
3852 007214 104423                      TWAT16      ;WAIT FOR 1ST INTERRUPT
3853 007216 104002                      ERROR    2      ;TO SLOW/NOT COMPLETE ERROR
3854
3855 007220 005037 001662                      CLR    INTSET      ;CLEAR INTERRUPT FLAG
3856
3857 007224 104437                      TWAT8S      ;WAIT FOR INTERRUPT
3858 007226 104002                      ERROR    2
3859
3860 007230 012737 007236 001110      MOV    #1$,SLPERR      ;SET LOCAL LOOP ON ERROR
3861
3862 007236 104416                      1$: TSSINIT      ;CLEAR SUBSYSTEM
3863 007240 104003                      ERROR    3      ;BAD INIT ERROR
3864
3865 007242 013737 001626 001610      MOV    DRVNUM,L.CS2      ;LOAD DRIVE NUMBER
3866 007250 012737 000117 001600      MOV    #SEEK,L.CS1      ;LOAD SEEK
3867 007256 110137 001607                      MOV    R1,L.DT      ;LOAD TRACK
3868
3869 007262 104417                      TLOADRK      ;LOAD RK REGS
3870 007264 104423                      TWAT16      ;WAIT FOR INTERRUPT
3871 007266 104002                      ERROR    2      ;TO SLOW/NOT COMPLETE
3872
3873 007270 032701 000001                      BIT    #BIT0,R1      ;TEST IF HEAD ADDRESS HAS BIT 0
3874 007274 001403                      BEQ     2$      ;NO - SKIP
3875 007276 032701 000002                      BIT    #BIT1,R1      ;TEST IF HEAD ADDRESS HAS BOTH 0 AND 1
3876 007302 001007                      BNE     3$      ;YES-GO CHECK BOTH IDAE AND SKI SET
3877
3878 007304 104422                      2$: TCHKWE      ;CHECK OPERATION WITH ERROR
3879 007306 002000                      IDAERR      ;ILLEGAL DISK ADDRESS ERROR
3880 007310 000000                      0
3881 007312 000000                      0
3882 007314 104004                      ERROR    4 ; OR 5,6,7 ;REPORT ALL DISCREPANCIES
3883 007316 104415                      SCOP1      ;LOCAL LOOP ON ERROR
3884 007320 000406                      BR      4$
3885
3886 007322 104422                      3$: TCHKWE      ;CHECK OPERATION WITH ERROR
3887 007324 002002                      IDAERR!SKIERR ;ILLEGAL DISK ADDRESS ERROR
3888 007326 000000                      0      ;SEEK INCOMPLETE
3889 007330 000000                      0
3890 007332 104004                      ERROR    4 ;OR 5,6,7 ;REPORT ANY DISCREPANCIES
3891 007334 104415                      SCOP1      ;LOCAL LOOP ON ERROR TO 1$
3892
3893 007336 005201 000010                      4$: INC    R1      ;ELSE BUMP TO NEXT ILLEGAL TRACK
3894 007340 022701                      CMP     #8.,R1      ;ALL ILLEGAL TRACKS SELECTED?
3895 007344 001334                      BNE     1$      ;NO-LOOP
3896
3897 ;*****
3898 ;TEST 17 ILLEGAL DISK ADDRESS (PART 2)
3899 ;
3900 ;
3901 ;ISSUE A SEEK TO CYLINDER 1000, HEAD 0. MAKE SURE
;ILLEGAL DISK ADDRESS ERROR SETS. CLEAR CONTROLLER AND DRIVE
;

```

M06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 76
T17 ILLEGAL DISK ADDRESS (PART 2)

SEQ 0076

```

3902
3903 007346 000004
3904 007350 012737 000012 001262
3905 007356 104416
3906 007360 104003
3907
3908 007362 012737 000113 001600
3909 007370 013737 001626 001610
3910
3911 007376 104417
3912
3913 007400 104423
3914 007402 104002
3915
3916 007404 005037 001662
3917
3918 007410 104437
3919 007412 104002
3920
3921 007414 012737 007422 001110
3922
3923 007422 104416
3924 007424 104003
3925
3926 007426 013737 001626 001610
3927 007434 012737 000117 001600
3928 007442 012737 001000 001614
3929
3930 007450 104417
3931 007452 104423
3932 007454 104002
3933
3934 007456 104422
3935 007460 002000
3936 007462 000000
3937 007464 000000
3938 007466 104004
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957

```

```

*****
1ST17: SCOPE
MOV #10., $TIMES ;DO 10. 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 #15, $LPERR ;SET LOOP TO BYPASS RECAL

15: 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 1DAERR ;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(B) 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 PACK 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.
*
*****
1ST20: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
CLRB 2$ ;CLEAR SECTOR POINTER
CLR R0
CLR R5 ;CLEAR R5 FOR BAD SECTOR COUNTING
MOV BSF26P,R3 ;SET POINT IN TO STORE BS 26 SECT FORMAT
MOV #1$,SLPERR ;SET ERROR RETURN ADDRESS FOR INTERNAL LOOP
1$: TSSINIT
ERROR 3 ;BAD INIT ERROR

JSR R4,LRLOAD ;LOAD "L" REGS WITH
RDATA ;READ DATA
-400 ;WORD COUNT
IBUFF ;BUFFER ADDRESS
2$: .BYTE 0 ;SECTOR ADDRESS
.BYTE 2 ;TRACK ADDRESS
632 ;CYLINDER ADDRESS

TLOADRK ;LOAD "L" REGS INTO RK
TWAT112 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;CHECK FOR ANY ERRORS
ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
SCOPI ;LOOP TO 1$ IF SW 9 SET

TSTB $ERFLG ;TEST FOR ERROR IN OPERATION
BEQ 7$ ;NO-SKIP
TST R0 ;GETTING A BS FACTORY SECTOR 26 SECT FORMAT?
BNE 3$ ;NO-SKIP
ADD #2,2$ ;NEXT SECTOR ADDRESS
CMPB #10.,2$ ;PAST APPLICABLE SECTORS?
BNE 6$ ;NO-SKIP
MOV #EMS1,EMIN
ERROR 1 ;"CANNOT READ BS FILES
25$: BIC DRVBIT,$DEVN ;CLEAR DRIVE FROM DRIVE MAP
BIS DRVBIT,DRVDRP ;SET DRIVE DROPPED
JMP NEWDRV ;ABORT TEST PASS.

3$: CMP #1,R0 ;GETTING A BS SOFT SECTOR 26 SECT FORMAT?
BNE 4$ ;NO-SKIP
ADD #2,2$ ;NEXT SECTOR ADDRESS
CMPB #22.,2$ ;PAST APPLICABLE SECTORS?
BNE 6$ ;NO-SKIP
MOV #EMS1,EMIN
ERROR 1 ;"CANNOT READ BS FILES"
BR 25$

4$: CMP #2,R0 ;GETTING A BS FACT SECTOR 24 SECTOR FORMAT?
BNE 5$ ;NO-SKIP
ADD #2,2$ ;NEXT SECTOR ADDRESS

```

B07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 78
T20 READ BAD SECTOR INFORMATION

SEQ 0078

4014	007712	122737	000013	007540		CMPB	#11.,2\$		;PAST APPLICABLE SECTORS?
4015	007720	001021				BNE	6\$		;NO-SKIP
4016									
4017	007722	012737	055734	001360		MOV	#EMS1,EMIN		
4018	007730	104001				ERROR	1		;"CANNOT READ BS FILES"
4019	007732	000732				BR	25\$		
4020									
4021	007734	062737	000002	007540	5\$:	ADD	#2,2\$		;NEXT SECTOR (BS SOFT 24 SECT MODE)
4022	007742	122737	000027	007540		CMPB	#23.,2\$		;PAST APPLICABLE SECTORS?
4023	007750	001005				BNE	6\$		;NO-SKIP
4024	007752	012737	055734	001360		MOV	#EMS1,EMIN		
4025	007760	104001				ERROR	1		;"CANNOT READ BS FILES"
4026	007762	000716				BR	25\$		
4027									
4028	007764	105037	001103		6\$:	CLRB	\$ERFLG		;CLEAR ERROR FLAG
4029	007770	000654				BR	1\$		;DO NEXT READ
4030									
4031	007772	005737	061422		7\$:	TST	IBUFF+6		;CHECK FOR ALIGNMENT PACK
4032	007776	001405				BEQ	8\$		;NO-SKIP
4033	010000	012737	056027	001360		MOV	#EMS2,EMIN		
4034	010006	104001				ERROR	1		;"ALIGNMENT PACK. DRIVE ABORTING"
4035	010010	000703				BR	25\$		
4036									
4037	010012	012701	061424		8\$:	MOV	#IBUFF+10,R1		;SET TO START OF BAD SECTOR DATA
4038									
4039	010016	022711	177777		9\$:	CMP	#-1,(R1)		;TEST IF WORD ALL ONES (END OF DATA)
4040	010022	001422				BEQ	11\$		;YES-SKIP
4041	010024	012123				MOV	(R1)+,(R3)+		;STORE CYLINDER
4042	010026	012123				MOV	(R1)+,(R3)+		;TRACK AND SECTOR
4043	010030	005205				INC	R5		;BUMP ERROR COUNTER
4044	010032	022705	000025			CMP	#21.,R5		;DOES IT TOTAL 20 FOR THIS FORMAT?
4045	010036	001367				BNE	9\$		;NO-TEST AND MORE NEXT ADDRESS
4046	010040	012737	056125	001360		MOV	#EMS3,EMIN		
4047	010046	104001				ERROR	1		;TO MANY BAD SECTORS
4048	010050	043737	001630	001354	10\$:	BIC	DRVBIT,\$DEVN		;CLEAR DRIVE FROM TESTING
4049	010056	053737	001630	001666		BIS	DRVBIT,DRVDRP		;SET DRIVE DROPPED
4050	010064	000137	027764			JMP	NEWDRV		;ABORT PASS
4051									
4052	010070	005200			11\$:	INC	R0		;BUMP TO NEXT
4053	010072	022700	000001			CMP	#1,R0		;NOW TESTING BS SOFT 26 SECTOR MODE?
4054	010076	001011				BNE	12\$		;NO-SKIP
4055	010100	012723	177777			MOV	#-1,(R3)+		;INSERT END OF FIELD FLAG
4056	010104	010337	001644			MOV	R3,\$S26P		;SET POINTER TO BAD SECTOR SOFTWARE FIELD
4057	010110	112737	000012	007540		MOVB	#12,2\$		;SET TO FIRST SECTOR THIS MODE
4058	010116	000137	007522			JMP	1\$		;GO READ IT.
4059	010122	022700	000002		12\$:	CMP	#2,R0		;NOW TESTING BS FACT 24 SECTOR MODE?
4060	010126	001014				BNE	13\$		;NO-SKIP
4061	010130	012723	177777			MOV	#-1,(R3)+		;INSERT END OF FIELD FLAG
4062	010134	112737	000001	007540		MOVB	#1,2\$		;SET TO FIRST SECTOR THIS MODE
4063	010142	010537	001646			MOV	R5,\$S26CT		;STORE TOTAL BS COUNT 26 SECTOR MODE
4064	010146	005005				CLR	R5		;CLEAR COUNTER FOR COUNTING 24 SECT BS
4065	010150	013703	001636			MOV	\$SF24P,R3		;SET POINTER FOR STORING BS
4066	010154	000137	007522			JMP	1\$		;GO READ
4067									
4068	010160	022700	000003		13\$:	CMP	#3,R0		;NOW TESTING BS SOFT 24 SECTOR MODE
4069	010164	001011				BNE	14\$		;NO-SKIP

C07

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZR6KD.P11 31-JAN-77 16:26

 MACY11 27(1006) 31-JAN-77 16:34 PAGE 79
 T20 READ BAD SECTOR INFORMATION

SEQ 0079

4070	010166	012723	177777			MOV	#-1,(R3)+	; INSERT END OF FIELD FLAG
4071	010172	010337	001642			MOV	R3,BSS24P	; STORE POINTER TO BSS 24 SECTOR MODE
4072	010176	112737	000013	007540		MOVB	#13,25	; GET START OF FIELDS BSS 24 SECT MODE
4073	010204	000137	007522			JMP	15	; GO READ IT
4074								
4075	010210	012723	177777		145:	MOV	#-1,(R3)+	; INSERT END OF FIELD FLAG
4076	010214	010537	001650			MOV	R5,BSS24CT	; STORE COUNT BSS 24 SECTOR MODE
4077								
4078								
4079	010220	012700	061042			MOV	#BS26,R0	; GET START OF BAD SECTOR BUFFER
4080	010224	012703	061252			MOV	#BS24+52.,R3	; GET END OF BUFFER
4081								
4082	010230	022720	000312		275:	CMP	#312,(R0)+	; TEST IF ANY SECTORS BAD IN CYL 312
4083	010234	001403				BEQ	265	; YES - GET OUT
4084								
4085	010236	020003				CMP	R0,R3	; CHECK IF ALL OF BUFFER TESTED
4086	010240	001373				BNE	275	; NO - LOOP
4087	010242	000406				BR	285	; EXIT
4088								
4089	010244	012737	057006	001360	265:	MOV	#DRVABT,EMIN	; "BAD SECTOR IN AREA FOR TEST"
4090	010252	104001				ERROR	1	
4091	010254	000137	007620			JMP	255	
4092								
4093	010260				285:			
4094								
4095								
4096								
4097								
4098								
4099								
4100								
4101								
4102								
4103								
4104	010260	000004						
4105	010262	012737	000012	001262		ST21: SCOPE		
4106	010270	012737	000312	001676		MOV	#10.,STIMES	; DO 10. ITERATIONS
4107	010276	105037	010421			MOV	#312,REFMT	; SET REFORMAT SWITCH
4108	010302	105037	010325			CLRB	105	; CLEAR SECTOR POINTER
4109	010306	104416				CLRB	115	; CLEAR TRACK POINTER
4110	010310	104003				TSSINIT		; CLEAR SUBSYSTEM
4111						ERROR	3	; BAD INIT ERROR
4112	010312	004437	035530		95:	JSR	R4,LRLoad	; LOAD "L" REGS
4113	010316	000127				WRHEAD		; WRITE HEADER
4114	010320	177676				-102		; WORD CNT FOR 26 SECTOR MODE
4115	010322	063414				0BUFF		; BUFFER
4116	010324	000				.BYTE	0	; SECTOR 0
4117	010325	000			115:	.BYTE	0	; TRACK 0
4118	010326	000312				312		; CYL 0
4119								
4120	010330	004437	041740			JSR	R4,GENCOM	; GENERATE DATA
4121	010334	000600				000600		; BUILD HEADERS-NO BAD SECTORS
4122	010336	012737	010346	001110		MOV	#1115,\$LPERR	; SET LOCAL LOOP ON ERROR
4123	010344	000402				BR	15	; SKIP INIT
4124	010346				1115:			
4125	010346	104416				TSSINIT		; CLEAR SUBSYSTEM

D07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 80
T21 FORMAT IN 26 SECTOR FORMAT

SEQ 0080

4126	010350	104003				ERROR 3	;BAD INIT ERROR
4127	010352	104417				TLOADRK	;LOAD RK REGS
4128	010354	104431				TWAT112	;WAIT FOR INTERRUPT
4129	010356	104002				ERROR 2	;TO SLOW/NOT COMPLETE ERROR
4130							
4131	010360	104421				TCHKOP	;CHECK FOR ANY ERRORS
4132	010362	104004				ERROR 4 ; OR 5,6,7	;REPORT ALL ERRORS
4133							
4134	010364	104415				SCOP1	;INTERNAL LOOP TO 1115
4135							
4136	010366	012737	010426	001110		MOV #112\$,SLPERR	;SET LOCAL LOOP ON ERROR
4137	010374	010203				MOV R2,R3	;BUILD POINTER TO RKDB
4138	010376	062703	000024			ADD #RKDB,R3	
4139	010402	012701	061414			MOV #IBUFF,R1	;SET POINTER TO BUFFER
4140	010406	004437	035530			JSR R4,LRLoad	;LOAD "L" REGS
4141	010412	000125				RDHEAD	;READ HEADER
4142	010414	000000				0	;NO WORDS COUNT
4143	010416	000000				0	;NO BUFFER
4144	010420	000				.BYTE 0	;SECTOR 0
4145	010421	000				.BYTE 0	;TRACK 0
4146	010422	000312				312	;CYL 312
4147							
4148	010424	000402				BR 25	;SKIP INIT
4149	010426						
4150	010426	104416				TSSINIT	;CLEAR SUBSYSTEM
4151	010430	104003				ERROR 3	;BAD INIT ERROR
4152	010432	104417				TLOADRK	;LOAD RK REGS
4153	010434	104423				TWAT16	;WAIT FOR INTERRUPT
4154	010436	104002				ERROR 2	;TO SLOW/NOT COMPLETE ERROR
4155							
4156	010440	104421				TCHKOP	;CHECK FOR ANY ERRORS
4157	010442	104004				ERROR 4; OR 5,6,7	;REPORT ALL ERRORS
4158	010444	012700	000003			MOV #3,R0	;SET COUNT
4159	010450	011321				MOV (R3),(R1)+	;GET RKDB
4160	010452	104420				TGETRK	;GET RK REGS
4161	010454	032737	100000	001550		BIT #DLT,T.CS2	;TEST IF DATA LATE
4162	010462	001410				BEQ 35	;NO-SKIP
4163	010464	012737	054527	054156		MOV #EMDLT,EM13	
4164	010472	012737	056675	061010		MOV #EMROB,DF011A	
4165	010500	104013				ERROR 13	; "DATA LATE SET RESULT OF DB READ
4166	010502	104415				SCOP1	;LOCAL LOOP TO 1125
4167							
4168	010504	032737	100000	001540		BIT #CERR,T.CS1	;TEST IF CONT ERROR SET
4169	010512	001410				BEQ 45	;NO-SKIP
4170	010514	012737	056721	001500		MOV #EMCERR,EM13N	
4171	010522	012737	056675	061010		MOV #EMROB,DF011A	
4172	010530	104013				ERROR 13	; "CERR SET RESULT OF READ DB
4173	010532	104415				SCOP1	;LOCAL LOOP TO 1125
4174	010534	005300				DEC R0	;DEC COUNT
4175	010536	001344				BNE 55	;LOOP IF NOT ZERO
4176	010540	012737	010550	001110		MOV #117\$,SLPERR	;SET LOCAL LOOP 1175
4177	010546	000402				BR 75	;SKIP INIT
4178	010550						
4179	010550	104416				TSSINIT	;CLEAR SUBSYSTEM
4180	010552	104003				ERROR 3	;BAD INIT ERROR
4181	010554	004437	036050			JSR R4,RDSTHD	;GO READ & SEQUENCE HEADERS

E07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 81
T21 FORMAT IN 26 SECTOR FORMAT

SEQ 0081

```

4182 010560 104421          TCHKOP          ;CONTROLLER ERROR RETURN
4183 010562 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4184 010564 104013          ERROR 13          ;"DATA LATE SET RESULT OF DATA BUFFER READ"
4185 010566 104002          ERROR 2          ;"OPERATION TOO SLOW" MESSAGE
4186                                     ;OR "HEADER 0 NOT FOUND" MESSAGE
4187
4188 010570 004437 041740    JSR      R4,GENCOM
4189 010574 100200          100200          ;COMPARE IBUF & OBUF (HEADERS)
4190 010576 000414          BR      6$      ;GOOD RETURN-NO MISCOMPARES
4191 010600 104015          ERROR 15          ;REPORT 1ST MISCOMPARES
4192
4193 010602 013700 001634    12$: MOV     ERRMT,R0      ;GET ERROR LIMIT
4194 010606 005300          DEC     R0        ;DECREMENT IT
4195 010610 001407          BEQ     6$        ;EXIT IF ZERO
4196 010612 004437 041740    JSR      R4,GENCOM
4197
4198 010616 040000          040000          ;RESUME COMPARE
4199 010620 000403          BR      6$      ;GOOD RETURN-NO MORE ERRORS
4200 010622 104016          ERROR 16          ;REPORT NEXT ERROR LINE
4201 010624 000770          BR      12$     ;LOOP
4202 010626 104415          SCOP1          ;LOCAL ERROR LOOP TO 117$
4203
4204 010630 105737 001607    6$: TSTB    L,DT        ;WAS TRACK 1 JUST DONE?
4205 010634 001010          BNE     8$      ;YES-SKIP
4206
4207 010636 112737 000001 010325 MOVB    #1,11$    ;CHANGE PARAM TO LOAD "L" WITH
4208 010644 112737 000001 010421 MOVB    #1,10$    ;TRACK 2
4209 010652 000137 010312    JMP      9$      ;JUMP TO DO ENTIRE TEST ON TRK 1
4210
4211 010656          8$:
4212
4213 .SBTTL **HEADER RECOGNITION TESTS
4214
4215 *****
4216 *TEST 22      BAD SECTOR ERROR
4217 *
4218 *      FORMAT CYLINDER 312, TRACK 0, ON SCRATCH PACK TO HAVE
4219 *      SECTOR 0 (BIT 15 OR WORD 2 OF HEADER RESET) AND SECTOR 1
4220 *      (BIT 14 OR WORD 2 OF HEADER RESET) TO BE BAD SECTORS
4221 *      AND ALL OTHER SECTORS GOOD.
4222 *
4223 *      ISSUE A WRITE DATA OR 400 WORDS TO CYLINDER 312, TRACK 0,
4224 *      SECTOR 0. MAKE SURE BAD SECTOR ERROR SETS. ISSUE A
4225 *      WRITE DATA TO CYLINDER 0, TRACK 0, SECTOR 1 OF 400 WORDS.
4226 *      MAKE SURE BAD SECTOR ERROR SET. ISSUE A WRITE DATA
4227 *      OF 400 WORDS TO CYLINDER 0, TRACK 0, SECTOR 2. MAKE
4228 *      SURE NO ERROR SETS.
4229 *
4230 *****
4231 010656 000004          TST2: SCOPE
4232 010660 012737 000012 001262 MOV     #10,$TIMES    ;DO 10. ITERATIONS
4233 010666 012737 000312 001676 MOV     #312,REFMT    ;SET REFORMAT SWITCH
4234 010674 104416          TSSINIT          ;CLEAR SUBSYSTEM
4235 010676 104003          ERROR 3          ;BAD INIT ERROR
4236
4237 010700 004437 035530    JSR      R4,LRL040 ;LOAD "L" REGS

```

F07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 82
 DZR6KD.P11 31-JAN-77 16:26 T22 BAD SECTOR ERROR

SEQ 0082

4238	010704	000127			WRHEAD		;WRITE HEADER
4239	010706	177676			-102		;WORD COUNT FOR 26 SECTOR MODE
4240	010710	063414			OBUFF		;BUFFER ADDRESS
4241	010712	000			.BYTE	0	;SECTOR
4242	010713	000			.BYTE	0	;TRACK
4243	010714	000312			312		;CYLINDER
4244							
4245							
4246	010716	004437	041740		JSR	R4,GENCOM	;GENERATE HEADERS
4247	010722	000600			600		;WITH NO BS BITS
4248							
4249	010724	012700	063416		MOV	#OBUFF+2,R0	;RESET BIT 15 IN WORD 2 OF
4250	010730	042720	100000		BIC	#BIT15,(R0)+	;SECTOR 0 HEADER AND BIT 14
4251	010734	042720	100000		BIC	#BIT15,(R0)+	;IN WORD 2 OF SECTOR 1 HEADER.
4252	010740	005720			TST	(R0)+	;ALSO CORRECT THE VRC
4253	010742	042720	040000		BIC	#BIT14,(R0)+	
4254	010746	042710	040000		BIC	#BIT14,(R0)	
4255							
4256	010752	104417			TLOADRK		;LOAD RK REGS
4257	010754	104431			TWAIT112		;WAIT FOR INTERRUPT
4258	010756	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4259							
4260	010760	104421			TCHKOP		;CHECK IF ANY ERRORS
4261	010762	104004			ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS
4262	010764	012737	010772	001110	MOV	#4\$,SLPERR	;SET LOCAL LOOP ON ERROR
4263	010772	104416		4\$:	TSSINIT		
4264	010774	104003			ERROR	3	
4265	010776	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
4266	011002	000123			WRDATA		;WRITE DATA
4267	011004	177400			-400		;WORD COUNT
4268	011006	063414			OBUFF		;BUS ADDRESS
4269	011010	000		5\$:	.BYTE	0	;SECT 0
4270	011011	000			.BYTE	0	;TRACK 0
4271	011012	000312			312		;CYL 312
4272							
4273	011014	104417		1\$:	TLOADRK		;LOAD RK REGS
4274	011016	104424			TWAIT32		;WAIT FOR INTERRUPT
4275	011020	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4276							
4277	011022	022737	000002	011010	CMP	#2,5\$	;JUST READ SECTOR 2?
4278	011030	001415			BEQ	6\$	;YES - SKIP
4279							
4280	011032	104422			TCHKWE		;CHECK OPERATION WITH ERROR
4281	011034	000000			0		
4282	011036	000100			100		;EXPECTED BSE
4283	011040	000000			0		
4284	011042	104004			ERROR	4 ; OR 5,6,7	;REPORT ANY DISCREPENCIES
4285							
4286	011044	104415			SCOP1		;LOCAL ERROR LOOP TO 4\$
4287							
4288	011046	122737	000002	011010	CMPB	#2,5\$	;WAS SECTOR SET TO 2
4289	011054	001405			BEQ	7\$	;YES-SKIP
4290	011056	105237	011010		INCB	5\$	;BUMP TO NEXT SECTOR
4291	011062	000743			BR	4\$	;LOOP
4292							
4293	011064	104421		6\$:	TCHKOP		;CHECK FOR GOOD OPERATION

SEQ 0083

```

1ST23: SCOPE
MOV      #10.,$TIMES      ;DO 10. ITERATIONS
MOV      #312,REFMT       ;SET REFORMAT SWITCH
TSSINIT  ;CLEAR SUBSYSTEM
ERROR    3                ;BAD INIT ERROR

JSR      R4,LRLOAD        ;LOAD "L" REGS
WRHEAD   ;WRITE HEADER
-102     ;WORD COUNT
OBUFF    ;BUFF ADD
.BYTE    0                ;SECT
.BYTE    0                ;TRACK
312      ;CYL

JSR      R4,GENCOM        ;BUILD HEADERS NO BSE
600

MOV      #OBUFF+4,R0      ;GET ADDRESS OF VRC WORD
MOV      #BIT0,R3         ;SET FOR BIT CHANGE SELECT
15:      BIT      R3,(R0)  ;CHECK A VRC BIT
BEQ      25               ;SKIP IF ZERO
BIC      R3,(R0)         ;ELSE CLEAR IT
BR       35               ;SKIP
25:      BIS      R3,(R0)  ;IF ZERO SET IT
35:      ADD      #6,R0    ;BUMP TO NEXT VRC WORD

ASL      R3                ;SHIFT THE SELECT
BNE      15               ;IF BIT NOT SHIFTED OUT-LOOP

TLOADRK  ;LOAD RK REGS
TWAIT112 ;WAIT FOR INTERRUPT
ERROR    2                ;TO SLOW/NOT COMPLETE ERROR
TCHKOP   ;CHECK OPERATION COMPLETE
ERROR    4 ; OR 5,6,7     ;REPORT ALL ERRORS

45:      MOV      #45,$LPERR ;SET LOCAL LOOP
TSSINIT  ;CLEAR SUBSYSTEM
ERROR    3                ;BAD INIT ERROR

JSR      R4,LRLOAD        ;LOAD "L" REGS
WRDATA   ;WRITE DATA
-400     ;WORD COUNT
OBUFF    ;BUFFER ADD

```

H07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 84
T23 HEADER VRC ERROR

SEQ 0084

```

4350 011226 000          55: .BYTE 0          ;SECT
4351 011227 000          .BYTE 0          ;TRACK
4352 011230 000312      312          ;CYL
4353
4354 011232 104417      TLOADRK          ;LOAD RK REG
4355 011234 104424      TWAT32          ;WAIT FOR INTERRUPT
4356 011236 104002      ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
4357
4358 011240 022737 000020 011226      CMP #16.,55 ;WAS THIS WRITE SECTOR 16?
4359 011246 001415      BEQ 65          ;YES-SKIP
4360
4361 011250 104422      TCHKWE          ;ELSE
4362 011252 000000      0              ;CHECK OPERATION WITH ERROR
4363 011254 000040      40              ;HVRC EM EXPECTED
4364 011256 000000      0
4365 011260 104004      ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
4366
4367 011262 104415      SCOP1          ;LOCAL LOOP TO 45
4368
4369 011264 105237 011226      INCB 55          ;BUMP SECTOR IN "L" REG
4370 011270 022737 000016 011226      CMP #16,55 ;IF SECTOR IS 16 OR LESS
4371 011276 003744      BLE 45          ;LOOP
4372 011300 000402      BR 75          ;ELSE EXIT
4373 011302 104421      65: TCHKOP          ;CHECK LAST OPERATION NO ERRORS
4374 011304 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4375
4376 011306      75:
4377      ;*****
4378      ;TEST 24 BAD SECTOR ERROR AND HVRC ERROR
4379      ;
4380      ; FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR ZERO HAS
4381      ; BOTH A BAD SECTOR ERROR AND HEADER VRC. ISSUE A WRITE DATA
4382      ; TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE ONLY HEADER VRC
4383      ; ERROR SETS.
4384      ;
4385      ;*****
4386 011306 000004      TEST24: SCOPE
4387 011310 012737 000012 001262      MOV #10.,STIMES ;DO 10. ITERATIONS
4388 011316 012737 000312 001676      MOV #312,REFMT ;SET REFORMAT SWITCH
4389 011324 104416      TSSINIT          ;CLEAR SUBSYSTEM
4390 011326 104003      ERROR 3          ;BAD INIT ERROR
4391
4392 011330 004437 035530      JSR R4,LRLOAD ;LOAD "L" REG
4393 011334 000127      WRHEAD          ;WRITE HEADER
4394 011336 177676      -102          ;WORD CNT FOR 26 SECTOR MODE
4395 011340 063414      OBUFF          ;BUFF ADD
4396 011342 000          .BYTE 0          ;SECTOR
4397 011343 000          .BYTE 0          ;TRACK
4398 011344 000312      312          ;CYLINDER
4399
4400 011346 004437 041740      JSR R4,GENCOM ;BUILD HEADERS-NO BSE
4401 011352 000600      600
4402
4403 011354 042737 100000 063416      BIC #BIT15,OBUFF+2 ;CLEAR BIT TO SET BSE,LEAVE VRC BAD.
4404
4405 011362 104417      TLOADRK          ;LOAD RK REGS

```

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRSKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 85
T24 BAD SECTOR ERROR AND HVRC ERROR

SEQ 0085

```

#06 011364 104431          TWTAT112          ;WAIT FOR INTERRUPT
#07 011366 104002          ERROR 2             ;TO SLOW/NOT COMPLETE ERROR
#08
#09 011370 104421          TCHKOP              ;CHECK FOR ANY ERRORS
#10 011372 104004          ERROR 4 ; OR 5,6,7   ;REPORT ALL ERRORS
#11
#12 011374 004437 035530   JSR R4,LRLOAD      ;LOAD "L" REGS
#13 011400 000123          WRDATA              ;WRITE DATA
#14 011402 177400          -400                ;WORD COUNT
#15 011404 063414          OBUFF              ;BUFF ADD
#16 011406 000            .BYTE 0             ;SECTOR
#17 011407 000            .BYTE 0             ;TRACK
#18 011410 000312          312                 ;CYLINDER
#19
#20 011412 104417          TLOADRK             ;LOAD RK REGS
#21 011414 104424          TWTAT32            ;WAIT FOR INTERRUPT
#22 011416 104002          ERROR 2             ;TO SLOW/NOT COMPLETE ERROR
#23
#24 011420 104422          TCHKWE              ;CHECK OPERATION WITH EXPECTED ERR
#25 011422 000000          0                   ;
#26 011424 000040          40                  ;HVRC ERR EXPECTED
#27 011426 000000          0                   ;
#28 011430 104004          ERROR 4 ; OR 5,6,7   ;REPORT ALL DISCREPENCIES
#29
#30 *****
#31 *TEST 25 OPERATION INCOMPLETE
#32 *
#33 *   FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 21 HAS THE
#34 *   WRONG FORMAT.  ISSUE A WRITE DATA OF 400 TO CYLINDER 0,
#35 *   TRACK 0, SECTOR 21.  MAKE SURE OPI SET.
#36 *
#37 *****
#38 ST25: SCOPE
#39 011432 000004          MOV #10,STIMES      ;DO 10. ITERATIONS
#40 011434 012737 000012 001262          MOV #312,REFMT ;SET REFORMAT SWITCH
#41 011442 012737 000312 001676          TSSINIT      ;CLEAR SUBSYSTEM
#42 011450 104416          ERROR 3             ;BAD INIT ERROR
#43 011452 104003
#44
#45 011454 004437 035530   JSR R4,LRLOAD      ;LOAD "L" REGS
#46 011460 000127          WRHEAD              ;WRITE HEADER
#47 011462 177676          -102                ;WORD COUNT FOR 26 SECT MODE
#48 011464 063414          OBUFF              ;BUFF ADD
#49 011466 000            .BYTE 0             ;SECTOR
#50 011467 000            .BYTE 0             ;TRACK
#51 011470 000312          312                 ;CYLINDER
#52
#53 011472 004437 041740   JSR R4,GENCOM
#54 011476 000600          600                 ;BUILD HEADERS-NO BSE ERRORS
#55
#56 011500 052737 001000 063614          BIS #BIT9,OBUFF+200 ;CHANGE FORMAT IN SECTOR 25
#57 011506 052737 001000 063616          BIS #BIT9,OBUFF+202 ;CORRECT THE VRC
#58
#59 011514 104417          TLOADRK             ;LOAD RK REGS
#60 011516 104431          TWTAT112            ;WAIT FOR INTERRUPT
#61 011520 104002          ERROR 2             ;TO SLOW/NOT COMPLETE

```

J07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 86
T25 OPERATION INCOMPLETE

SEQ 0086

```

4462 011522 104421      TCHKOP      ;CHECK FOR ANY ERRORS
4463 011524 104004      ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
4464
4465 011526 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGS
4466 011532 000123      WRDATA      ;WRITE DATA
4467 011534 177400      -400         ;400 WORDS
4468 011536 063414      OBUFF       ;BUFF ADDR
4469 011540 025         .BYTE 25     ;SECTOR 25
4470 011541 000         .BYTE 0      ;TRACK 0
4471 011542 000312      312         ;CYL 312
4472
4473 011544 104417      TLOADRK      ;LOAD RK REGS
4474 011546 104425      TWAT48      ;WAIT FOR INTERRUPT
4475 011550 104002      ERROR 2      ;TO SLOW/NOT COMPLETE
4476
4477 011552 104422      TCHKWE      ;CHECK OPERATION EXPECTED ERROR
4478 011554 000000      0           ;
4479 011556 000020      20          ;OPI EXPECTED
4480 011560 000000      0           ;
4481 011562 104004      ERROR 4 ; OR 5,6,7 ;REPORT ANY DISCREPENCIES
4482 *****
4483 *TEST 26 OPI WITH HVRC ERROR
4484 *
4485 *   FORMAT CYLINDER 312, TRACK 0 SUCH THAT A HEADER VRC
4486 *   ERROR IS PRESENT AND SECTOR 17 HAS THE WRONG FORMAT.
4487 *   ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312,
4488 *   TRACK 0, SECTOR 17. THAT BOTH OPERATION INCOMPLETE
4489 *   AND HEADER VRC SET.
4490 *
4491 *****
4492 011564 000004      ST26: SCOPE
4493 011566 012737 000012 001262 MOV #10,STIMES ;DO 10 ITERATIONS
4494 011574 012737 000312 001676 MOV #312,REFMT ;SET REFORMAT SWITCH
4495 011602 104416      TSSINIT      ;CLEAR SUBSYSTEM
4496 011604 104003      ERROR 3      ;BAD INIT ERROR
4497
4498 011606 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGS
4499 011612 000127      WRHEAD      ;WRITE HEADER
4500 011614 177676      -102        ;WORD COUNT FOR 26 SECT MODE
4501 011616 063414      OBUFF       ;BUS ADDRESS
4502 011620 000         .BYTE 0      ;SECTOR
4503 011621 000         .BYTE 0      ;TRACK
4504 011622 000312      312         ;CYLINDER
4505
4506 011624 004437 041740 JSR R4,GENCOM ;BUILD HEADER- NO BSE ERRORS
4507 011630 000600      600
4508
4509 011632 012700 063550 MOV #OBUFF+134,R0 ;GET ADDRESS 2ND WORD HDR 17(8)
4510 011636 052720 001000 BIS #BIT9,(R0)+ ;SET FORMAT 24 SECT PER TRACK
4511 011642 052720 001000 BIS #BIT9,(R0)+ ;SET VRC BIT
4512 011646 062700 000004 ADD #4,R0 ;BUMP TO HVRC WORD HDR 20(8)
4513 011652 032710 000001 BIT #BIT0,(R0) ;TEST BIT 0
4514 011656 001403      BEQ 1$      ;RESET-SKIP
4515 011660 042710 000001 BIC #BIT0,(R0) ;CLEAR BIT
4516 011664 000402      BR 2$
4517 011666 052710 000001 1$: BIS #BIT0,(R0) ;SET BIT

```

K07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 87
T26 OPI WITH HVRC ERROR

SEQ 0087

```

4518                                     ;FORCE OPI AND HVRC ERROR
4519 011672 104417 25: TLOADRK          ;LOAD RK REGS
4520 011674 104431 TWAT112          ;WAIT FOR INTERRUPT
4521 011676 104002 ERROR 2          ;TO SLOW/NOT COMPLETE ERROR
4522
4523 011700 104421 TCHKOP          ;CHECK FOR ANY ERRORS
4524 011702 104004 ERROR 4 ; OR 5,6,7 ;YES-REPORT ALL ERRORS
4525
4526 011704 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGS
4527 011710 000123 WRDATA          ;WRITE DATA
4528 011712 177400 -400           ;400 WORDS
4529 011714 063414 OBUF          ;BUFF ADDRESS
4530 011716 017 .BYTE 17          ;SECT 17
4531 011717 000 .BYTE 0          ;TRACK 0
4532 011720 000312 312           ;CYLINDER 312
4533
4534 011722 104417 TLOADRK          ;LOAD RK REGS
4535 011724 104425 TWAT48          ;WAIT FOR INTERRUPT
4536 011726 104002 ERROR 2          ;TO SLOW/NOT COMPLETE
4537
4538 011730 104422 TCHKWE          ;CHECK WITH EXPECTED ERROR
4539 011732 000000 0             ;
4540 011734 000060 60            ;HVRC ERR & OPI EXPECTED
4541 011736 000000 0             ;
4542 011740 104004 ERROR 4 ;OR 5,6,7 ;
4543 *****
4544 *TEST 27 HVRC IGNORE ON NON-ADDRESSED SECTOR
4545 *
4546 * FORMAT CYLINDER 312, TRACK 0 SUCH THAT SECTOR 20 HAS AN HVRC
4547 * ERROR. ISSUE A WRITE DATA OF 400 WORDS TO CYLINDER 312, TRACK 0,
4548 * AND SECTOR 21. MAKE SURE HVRC IS NOT SET AT THE
4549 * END OF THE OPERATION
4550 *
4551 *****
4552 011742 000004 ST27: SCOPE
4553 011744 012737 MOV #10, $TIMES ;DO 10. ITERATIONS
4554 011752 012737 MOV #312, REFM ;SET REFORMAT SWITCH
4555
4556 011760 104416 .NIT          ;CLEAR SUBSYSTEM
4557 011762 104003 ROR 3          ;BAD INIT ERROR
4558
4559 011764 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGISTERS
4560 011770 000127 WRHEAD          ;WRITE HEADER
4561 011772 177676 -102          ;WORD COUNT FOR 26 SECTOR MODE
4562 011774 063414 OBUF          ;BUFF ADD
4563 011776 000 .BYTE 0          ;SECTOR
4564 011777 000 .BYTE 0          ;TRACK
4565 012000 000312 312           ;CYLINDER
4566
4567 012002 004437 041740 JSR R4,GENCOM ;BUILD HEADERS-NO BSE ERRORS
4568 012006 000600 600
4569
4570 012010 012700 063560 MOV #OBUF+144, R0 ;ADDRESS OF HEAD 20 HVRC WORD
4571 012014 012701 000002 MOV #BIT1, R1 ;BIT 1 CONSTANT
4572 012020 030110 BIT R1, (R0) ;TEST BIT 1 SET
4573 012022 001402 BEQ 15 ;RESET-SKIP

```

L07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 88
T27 HVRC IGNORE ON NON-ADDRESSED SECTOR

SEQ 0088

4574	012024	040110		BIC	R1,(R0)	;ELSE CLEAR BIT 1
4575	012026	000401		BR	2\$	;SKIP
4576	012030	050110		1\$: BIS	R1,(R0)	;SET BIT 1
4577						
4578	012032	104417		2\$: TLOADRK		;LOAD RK REGS
4579	012034	104431		TWAT112		;WAIT FOR INTERRUPT
4580	012036	104002		ERROR	2	;TO SLOW/NOT COMPLETE
4581						
4582	012040	104421		TCHKOP		;CHECK FOR ANY ERROR
4583	012042	104004		ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS
4584						
4585	012044	004437	035530	JSR	R4,LRLOAD	;LOAD "L" REGISTER
4586	012050	000123		WRDATA		;WRITE DATA
4587	012052	177400		-400		;WORD COUNT
4588	012054	063414		OBUFF		;BUFF ADD
4589	012056	021		.BYTE	21	;SECTOR
4590	012057	000		.BYTE	0	;TRACK
4591	012060	000312		312		;CYLINDER
4592	012062	104417		TLOADRK		;LOAD RK REGS
4593	012064	104424		TWAT32		;WAIT FOR INTERRUPT
4594	012066	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4595						
4596	012070	104421		TCHKOP		;CHECK FOR ANY ERROR
4597	012072	104004		ERROR	4 ; OR 5,6,7	;REPORT ALL ERRORS.
4598						
4599						
4600						

.SBTTL **DATA TRANSFER TESTS

```

*****
:TEST 30      WRITE AND READ ONE SECTOR
:
:      FORMAT CYLINDER 312, ALL TRACKS AND CYLINDER 313, TRACK 0
:      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.
:
*****

```

4611	012074	000004		†ST30: SCOPE		
4612	012076	012737	000012 001262	MOV	#10,\$TIMES	;DO 10. ITERATIONS
4613	012104	012737	000312 001676	MOV	#312,REFMT	;SET REFORMAT SWITCH
4614	012112	104416		TSSINIT		;CLEAR SUBSYSTEM
4615	012114	104003		ERROR	3	;BAD INIT ERROR
4616						
4617	012116	012737	000312 012144	MOV	#312,7\$	;PRESET CYL POINTER
4618	012124	105037	012143	CLRB	2\$	;CLEAR TRACK POINTER
4619						
4620	012130	004437	035530	1\$: JSR	R4,LRLOAD	;LOAD "L" REG
4621	012134	000127		WRHEAD		;WRITE HEADER
4622	012136	177676		-102		;WORD COUNT FOR 26 SECTOR MODE
4623	012140	063414		OBUFF		;BUFF ADDRESS
4624	012142	000		.BYTE	0	;SECTOR
4625	012143	000		.BYTE	0	;TRACK
4626	012144	000312		312		;CYLINDER
4627						
4628	012146	004437	041740	JSR	R4,GENCOM	
4629	012152	001200		1200		;BUILD HDRS-INCLUDE BAD SECTORS

M07

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 89
 DZR6KD.P11 31-JAN-77 16:26 T30 WRITE AND READ ONE SECTOR

SEQ 0089

4630											
4631	012154	104417				TLOADRK				;LOAD RK REGS	
4632	012156	104431				TWAT112				;WAIT FOR INTERRUPT	
4633	012160	104002				ERROR	2			;TO SLOW/NOT COMPLETE ERROR	
4634											
4635	012162	104421				TCHKOP				;CHECK FOR ANY ERRORS	
4636	012164	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS	
4637											
4638	012166	022737	000313	012144		CMP	#313,7\$			;TEST IF DONE 313 TK 0	
4639	012174	001414				BEQ	3\$			;YES - SKIP	
4640	012176	123727	012143	000002		CMPB	2\$, #2			;DID WE JUST FORMAT TRACK 2	
4641	012204	001403				BEQ	8\$			;YES-SKIP	
4642	012206	105237	012143			INCB	2\$			;BUMP TO NEXT TRACK	
4643	012212	000746				BR	1\$			;GO FORMAT NEXT TRACK	
4644											
4645	012214	105037	012143		8\$:	CLRB	2\$			;CLEAR TRACK POINTER	
4646	012220	005237	012144			INC	7\$			;BUMP CYL TO 313	
4647	012224	000741				BR	1\$			;GO FORMAT 313 TK 0	
4648											
4649	012226	004437	035530		3\$:	JSR	R4,LRLOAD			;LOAD "L" REGS	
4650	012232	000123				WRDATA				;WRITE DATA	
4651	012234	177400				-400				;ONE SECTOR WORD COUNT	
4652	012236	063414				0BUFF				;BUFF ADDRESS	
4653	012240	012				.BYTE	12			;SECTOR 12	
4654	012241	000				.BYTE	0			;TRACK 0	
4655	012242	000312				312				;CYLINDER 312	
4656											
4657	012244	004437	041740			JSR	R4,GENCOM				
4658	012250	000001				1				;BUILD DATA PATTERN 1	
4659	012252	000400				400				;400 WORDS LONG	
4660	012254	012737	012262	001110		MOV	#4\$,SLPERR			;SET FOR LOCAL LOOP	
4661	012262	104417			4\$:	TLOADRK				;LOAD RK REGS	
4662	012264	104431				TWAT112				;WAIT FOR INTERRUPT	
4663	012266	104002				ERROR	2			;TO SLOW/NOT COMPLETE ERROR	
4664											
4665	012270	104421				TCHKOP				;CHECK FOR ANY ERRORS	
4666	012272	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS	
4667											
4668	012274	004437	035530			JSR	R4,LRLOAD			;LOAD "L" REGS	
4669	012300	000121				RDDATA				;READ DATA	
4670	012302	177400				-400				;400 WORDS	
4671	012304	061414				IBUFF				;BUFF ADD	
4672	012306	012				.BYTE	12			;SECTOR 12	
4673	012307	000				.BYTE	0			;TRACK 0	
4674	012310	000312				312				;CYL 312	
4675											
4676	012312	104417				TLOADRK				;LOAD RK	
4677	012314	104424				TWAT32				;WAIT FOR INTERRUPT	
4678	012316	104002				ERROR	2			;TO SLOW/NOT COMPLETE	
4679											
4680	012320	104421				TCHKOP				;CHECK FOR ANY ERRORS	
4681	012322	104004				ERROR	4 ; OR 5,6,7			;REPORT ALL ERRORS	
4682											
4683	012324	004437	041740			JSR	R4,GENCOM				
4684	012330	100001				100001				;GO COMPARE DATA TO PATTERN 1	
4685	012332	000400				400				;400 WORDS LONG	

NO7

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 90
T30 WRITE AND READ ONE SECTOR

SEQ 0090

4686	012334	000413		BR	6\$	;GOOD RETURN-NO DATA ERRORS
4687	012336	104015		ERROR	15	;ERROR RETURN
4688						
4689	012340	013700	001634	MOV	ERRLMT,R0	;GET ERROR LIMIT
4690	012344	005300		SS: DEC	R0	;DEC LIMIT
4691	012346	001406		BEQ	6\$	;EXIT IF 0
4692	012350	004437	041740	JSR	R4,GENCOM	
4693	012354	040000		040000		;CONTINUE COMPARE
4694	012356	000402		BR	6\$	;EXIT IF NO MORE ERRORS
4695	012360	104016		ERROR	16	;ELSE REPORT MISCOMPARE
4696	012362	000770		BR	5\$	;LOOP
4697	012364	005037	001676	6\$: CLR	REFMT	;CLEAR REFORMAT SWITCH
4698						
4699						
4700						
4701						
4702						
4703						
4704						
4705						
4706						
4707						
4708	012370	000004				
4709	012372	012737	000012 001262	TST31: SCOPE		
4710				MOV	#10.,\$TIMES	;DO 10. ITERATIONS
4711	012400	104416		TSSINIT		;CLEAR SUBSYSTEM
4712	012402	104003		ERROR	3	;BAD INIT ERROR
4713						
4714	012404	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
4715	012410	000123		WRDATA		;WRDATA
4716	012412	177400		-400		; -400 WORDS
4717	012414	063414		OBUFF		;OBUFF IS BUFF ADDRESS
4718	012416	012		.BYTE	12	;SECTOR 12
4719	012417	001		.BYTE	1	;TRACK 1
4720	012420	000312		312		;CYLINDER 312
4721						
4722	012422	052737	000020 001610	BIS	#BAI,L.CS2	;SET INCREMENT INHIBIT
4723	012430	004437	041740	JSR	R4,GENCOM	;BUILD PATTERN
4724	012434	000016		16		;PATTERN 16
4725	012436	000400		400		;400 WORDS
4726						
4727	012440	104417		TLOWORK		;LOAD RK REGS
4728	012442	104430		TWAT96		;WAIT FOR INTERRUPT
4729	012444	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4730						
4731	012446	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
4732	012450	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
4733						
4734	012452	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
4735	012456	000121		RDDATA		;RDDATA
4736	012460	177400		-400		; -400 WORDS
4737	012462	061414		IBUFF		;IBUFF IS BUFF ADDRESS
4738	012464	012		.BYTE	12	;SECTOR 12
4739	012465	001		.BYTE	1	;TRACK 1
4740	012466	000312		312		;CYLINDER 312
4741						

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 91
T31 WRITE DATA WITH BUS ADDRESS INCREMENT INHIBIT

SEQ 0091

4742	012470	012700	000377		MOV	#377,R0	;SET COUNT TO SET OBUF TO BE
4743	012474	012701	063416		MOV	#OBUF+2,R1	;ALL THE FIRST WORD OF PATTERN
4744	012500	012703	063414		MOV	#OBUF,R3	
4745							
4746	012504	011321		15:	MOV	(R3),(R1)+	;MOV THE WORD
4747	012506	005300			DEC	R0	
4748	012510	001375			BNE	15	;LOOP UNTIL ALL WORDS SET
4749							
4750	012512	104417			TLOADRK		;LOAD RK REGS
4751	012514	104424			TWAIT32		;WAIT FOR INTERRUPT
4752	012516	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
4753							
4754	012520	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
4755	012522	104004			ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
4756							
4757	012524	004437	041740		JSR	R4,GENCOM	;COMPARE THE DATA
4758	012530	100000			100000		
4759	012532	000400			400		
4760	012534	000413			BR	25	
4761	012536	104015			ERROR	15	
4762	012540	013700	001634		MOV	ERRLMT,R0	;GET ERROR LIMIT
4763	012544	005300		645:	DEC	R0	;DECREMENT COUNT
4764	012546	001406			BEG	655	;IF ZERO - EXIT
4765	012550	004437	041740		JSR	R4,GENCOM	;CONTINUE DATA COMPARE
4766	012554	040000			40000		
4767	012556	000402			BR	655	;NO MORE ERRORS - EXIT
4768	012560	104016			ERROR	16	;REPORT NEXT ERROR
4769	012562	000770			BR	645	;LOOP
4770	012564			655:			
4771							
4772	012564			25:			
4773					*****		
4774					TEST 32	WRITE DATA ADDRESS GREATER THAN 32K	
4775					*		
4776					*	ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 177770.	
4777					*	MAKE SURE CORRECT DATA IS ON DISK.	
4778					*		
4779					*	NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 32K	
4780					*	OF MEMORY IS PRESENT.	
4781					*		
4782					*****		
4783	012564	000004		15:	SCOPE		
4784	012566	012737	000012 001262		MOV	#10, \$TIMES	;DO 10. ITERATIONS
4785	012574	123727	001326 000001		CMPB	\$MAMS1, #1	;TEST IF >32K MEM
4786	012602	002016			BGE	25	;YES-SKIP
4787							
4788	012604	032737	000002 001664		BIT	#MEMSZB, OPTFLG	;TEST IF REPORT ALREADY MADE
4789	012612	001011			BNE	15	;YES -SKIP
4790							
4791	012614	104401	051471		TYPE	,OPR011	;INSUFFICIENT MEMORY DATA TRANSFER WITH
4792	012620	104401	051634		TYPE	,OPR012	;ADDRESS >32K
4793	012624	052737	000002 001664		BIS	#MEMSZB, OPTFLG	;SET FLAG
4794	012632	104401	051650		TYPE	,OPR015	;BYPASSED
4795	012636	000470		15:	BR	45	;EXIT
4796	012640	012737	012646 001110	25:	MOV	#55, \$LPERR	;SET LOCAL LOOP ON ERROR ADDRESS
4797	012646			55:			

4798	012646	104416		TSSINIT		:CLEAR SUBSYSTEM
4799	012650	104003		ERROR 3		:BAD INIT ERROR
4800						
4801	012652	004437	035530	JSR R4,LRLoad		:LOAD "L" REGS
4802	012656	000123		WRDATA		:WRITE DATA
4803	012660	177400		-400		:400 WORDS
4804	012662	177770		177770		:BUS ADDRESS IN 32K -10 BYTES
4805	012664	016		.BYTE 16		:SECTOR 16
4806	012665	000		.BYTE 0		:TRACK 0
4807	012666	000312		312		:CYLINDER 312
4808	012670	004437	041740	JSR R4,GENCOM		:GENERATE DATA
4809	012674	010010		10010		:PATTERN 10, MEM. MANAGEMENT FOR DEST.
4810	012676	001777		1777		:RELOCATION ARGUMENT
4811	012700	000400		400		:400 WORDS
4812						
4813	012702	104417		TLOADRK		:LOAD RK REGS
4814	012704	104430		TWAIT96		:WAIT FOR INTERRUPT
4815	012706	104002		ERROR 2		:TO SLOW/NOT COMPLETE ERROR
4816						
4817	012710	104421		TCHKOP		:CHECK OPERATION FOR ANY ERRORS
4818	012712	104004		ERROR 4 ;OR 5, 6, 7,		:REPORT ALL ERRORS
4819	012714	104415		SCOP1		:LOCAL LOOP ON ERROR TO 55
4820						
4821	012716	004437	041740	JSR R4,GENCOM		:CLEAR Ibuff TO 1'S.
4822	012722	002007		2007		
4823	012724	001000		1000		
4824						
4825	012726	004437	035530	JSR R4,LRLoad		:LOAD "L" REGS
4826	012732	000121		RDDATA		:RDDATA
4827	012734	177400		-400		:400 WORDS
4828	012736	061414		IBUFF		:IBUFF IS BUFF ADDRESS
4829	012740	016		.BYTE 16		:SECTOR 16
4830	012741	000		.BYTE 0		:TRACK 0
4831	012742	000312		312		:CYLINDER 312
4832	012744	104417		TLOADRK		:LOAD RK REGS
4833	012746	104424		TWAIT32		:WAIT FOR INTERRUPT
4834	012750	104002		ERROR 2		:TO SLOW/NOT COMPLETE ERROR
4835	012752	104421		TCHKOP		:CHECK OPERATION FOR ANY ERRORS
4836	012754	104004		ERROR 4 ;OR 5, 6, 7,		:REPORT ALL ERRORS
4837	012756	004437	041740	JSR R4,GENCOM		:COMPARE
4838	012762	110000		110000		:MEMORY MANAGEMENT FOR DESTINATION
4839	012764	001777		1777		:RELOCATION ARGUMENT
4840	012766	000400		400		:400 WORDS
4841	012770	000413		BR 45		:NO ERROR-SKIP
4842	012772	104015		ERROR 15		:REPORT FIRST MISCOMPARE
4843	012774	013700	001634	MOV ERRLMT,R0		:GET ERROR LIMIT
4844	013000	005300		DEC R0		:DECREMENT COUNT
4845	013002	001406		BEQ 655		:IF ZERO - EXIT
4846	013004	004437	041740	JSR R4,GENCOM		:CONTINUE DATA COMPARE
4847	013010	050000		50000		
4848	013012	000402		BR 655		:NO MORE ERRORS - EXIT
4849	013014	104016		ERROR 16		:REPORT NEXT ERROR
4850	013016	000770		BR 645		:LOOP
4851	013020		655:			
4852						
4853	013020		45:			

D08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 93
T33 READ DATA ADDRESS GREATER THAN 32K

SEQ 0093

```

4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864 013020 000004
4865 013022 012737 000012 001262
4866 013030 123727 001326 000001
4867 013036 002001
4868
4869 013040 000462
4870
4871 013042 012737 013050 001110
4872
4873 013050
4874 013050 104416
4875 013052 104003
4876 013054 004437 035530
4877 013060 000123
4878 013062 177400
4879 013064 063414
4880 013066 017
4881 013067 000
4882 013070 000312
4883 013072 004437 041740
4884 013076 000011
4885 013100 000400
4886
4887 013102 104417
4888 013104 104430
4889 013106 104002
4890
4891 013110 104421
4892 013112 104004
4893 013114 004437 035530
4894 013120 000121
4895 013122 177400
4896 013124 177770
4897 013126 017
4898 013127 000
4899 013130 000312
4900
4901 013132 104417
4902 013134 104424
4903 013136 104002
4904 013140 104421
4905 013142 104004
4906 013144 004437 041740
4907 013150 120000
4908 013152 001777
4909 013154 000400

```

```

*****
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.
*****
T33: SCOPE
MOV #10,STIMES ;DO 10. ITERATIONS
CMPB $MAMS1,#1 ;CHECK IF >32K MEMORY
BGE 2$ ;YES-SKIP
1$: BR 5$ ;EXIT
2$: MOV #3$,SLPERR ;SET LOCAL ERROR LOOP
3$:
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-400 ;400 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 17 ;SECTOR 17
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312
JSR R4,GENCOM ;GENERATE DATA IN OBUFF
11 ;PATTERN 11
400 ;400 WORDS
TLOADRK ;LOAD RK REGS
TWAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;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
TWAT32 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKOP ;CHECK OPERATION FOR ANY ERRORS
ERROR 4 ;OR 5 6, 7, 10 ;REPORT ALL ERRORS
JSR R4,GENCOM ;COMPARE DATA
120000 ;MEMORY MANAGEMENT WITH SOURCE
1777 ;RELOCATION ARGUMENT
400 ;COMPARE 400 WORDS

```

E08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRSKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 94
T33 READ DATA ADDRESS GREATER THAN 32K

SEQ 0094

4910	013156	000413			BR	55		; NO MISCOMPARE-EXIT
4911	013160	104015			ERROR	15		; REPORT FIRST MISCOMPARE
4912	013162	013700	001634		MOV	ERRLMT, R0		; GET ERROR LIMIT
4913	013166	005300			645: DEC	R0		; DECREMENT COUNT
4914	013170	001406			BEQ	655		; IF ZERO - EXIT
4915	013172	004437	041740		JSR	R4, GENCOM		; CONTINUE DATA COMPARE
4916	013176	060000			60000			
4917	013200	000402			BR	655		; NO MORE ERRORS - EXIT
4918	013202	104016			ERROR	16		; REPORT NEXT ERROR
4919	013204	000770			BR	645		; LOOP
4920	013206				655:			
4921	013206				55:			
4922					*****			
4923					; TEST 34 WRITE DATA ADDRESS GREATER THAN 64K			
4924					; *			
4925					; ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 377770.			
4926					; MAKE SURE CORRECT DATA IS ON DISK.			
4927					; *			
4928					; NOTE: THIS TEST IS ONLY EXECUTED IF MORE THAN 64K			
4929					; OF MEMORY IS PRESENT.			
4930					; *			
4931					*****			
4932	013206	000004			TEST34:	SCOPE		
4933	013210	012737	000012	001262	MOV	#10, \$TIMES		; DO 10. ITERATIONS
4934	013216	123727	001326	000002	CMPB	\$MAMS1, #2		; CHECK IF >64K MEMORY
4935	013224	002016			BGE	25		; YES-SKIP
4936	013226	032737	000002	001664	65: BIT	#MEMSZB, OPTFLG		; TEST IF REPORT FLAG SET
4937	013234	001011			BNE	15		; NO-SKIP
4938								
4939	013236	104401	051471		TYPE	, OPR011		; "INSUFFICIENT MEMORY-DATA XFER WITH
4940	013242	104401	051640		TYPE	, OPR013		; ADDRESS >64K
4941	013246	104401	051650		TYPE	, OPR015		; BYPASSED"
4942	013252	052737	000002	001664	BIS	#MEMSZB, OPTFLG		; SET FLAG
4943	013260	000467			15: BR	55		
4944								
4945	013262	012737	013270	001110	25: MOV	#35, \$LPERR		; SET LOCAL LOOP ON ERROR
4946								
4947	013270				35:			
4948	013270	104416			TSSINIT			; CLEAR SUBSYSTEM
4949	013272	104003			ERROR	3		; BAD INIT ERROR
4950	013274	004437	041740		JSR	R4, GENCOM		; GENERATE DATA, PATTERN 11
4951	013300	010011			10011			; MEM MANAGEMENT ON DESTINATION
4952	013302	003777			3777			; RELOCATION ARGUMENT
4953	013304	000400			400			; 400 WORDS
4954								
4955	013306	004437	035530		JSR	R4, LRLOAD		; LOAD "L" REGS
4956	013312	000523			WRDATA: BA16			; WRITE DATA AND SET BA16
4957	013314	177400			-400			; 400 WORDS
4958	013316	177770			177770			; ACCROSS 64K BOUNDARY
4959	013320	020			.BYTE	20		; SECTOR 20
4960	013321	000			.BYTE	0		; TRACK 0
4961	013322	000312			312			; CYLINDER 312
4962								
4963	013324	104417			TLOADRK			; LOAD RK REGS
4964	013326	104430			TWAIT96			; WAIT FOR INTERRUPT
4965	013330	104002			ERROR	2		; TO SLOW/NOT COMPLETE ERROR

F08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 95
T34 WRITE DATA ADDRESS GREATER THAN 64K

SEQ 0095

```

4966
4967 013332 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
4968 013334 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4969 013336 004437 041740   JSR R4,GENCOM ;CLEAR Ibuff TO 1'S
4970 013342 002007          2007
4971 013344 001000          1000
4972
4973 013346 004437 035530   JSR R4,LRLOAD ;LOAD "L" REGS
4974 013352 000121          RDDATA ;RDDATA
4975 013354 177400          -400 ; -400 WORDS
4976 013356 061414          Ibuff ;IBUFF IS BUFF ADDRESS
4977 013360 020            .BYTE 20 ;SECTOR 20
4978 013361 000            .BYTE 0 ;TRACK 0
4979 013362 000312          312 ;CYLINDER 312
4980 013364 104417          TLOADRK ;LOAD RK REGS
4981 013366 104424          TWAT32 ;WAIT FOR INTERRUPT
4982 013370 104002          ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
4983
4984 013372 104421          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
4985 013374 104004          ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
4986 013376 004437 041740   JSR R4,GENCOM ;CHECK DATA
4987 013402 110000          110000 ;MEMORY MANAGEMENT WITH DESTINATION
4988 013404 003777          3777 ;RELOCATION ARGUMENT
4989 013406 000400          400 ;400 WORDS
4990 013410 000413          BR 5$ ;NO MISCOMPARES-SKIP
4991 013412 104015          ERROR 15 ;REPORT FIRST ERROR
4992
4993 013414 013700 001634   MOV ERLMT,R0 ;GET ERROR LIMIT
4994 013420 005300          64$: DEC R0 ;DECREMENT COUNT
4995 013422 001406          BEQ 65$ ;IF ZERO - EXIT
4996 013424 004437 041740   JSR R4,GENCOM ;CONTINUE DATA COMPARE
4997 013430 050000          50000
4998 013432 000402          BR 65$ ;NO MORE ERRORS - EXIT
4999 013434 104016          ERROR 16 ;REPORT NEXT ERROR
5000 013436 000770          BR 64$ ;LOOP
5001
5002 013440
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013 013440 000004          *****
5014 013442 012737 000012 001262  ;TEST 35 READ DATA ADDRESS GREATER THAN 64K
5015 013450 123727 001326 000002  ;
5016 013456 002001          ;
5017 013460 000462          ;
5018
5019 013462 012737 000032 001110  ;
5020
5021 013470          ;

```

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.

```

;ST35: SCOPE
MOV #10,$TIMES ;DO 10. ITERATIONS
CMPB $MAMS1,#2 ;CHECK IF >64K MEMORY
BGE 2$ ;YES-SKIP
BR 5$ ;EXIT
1$:
2$: MOV #32,$LPERR ;SET LOCAL LOOP ON ERROR
3$:

```

G08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 96
T35 READ DATA ADDRESS GREATER THAN 64K

SEQ 0096

5022	013470	104416		TSSINIT		;CLEAR SUBSYSTEM
5023	013472	104303		ERROR 3		;BAD INIT ERROR
5024	013474	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
5025	013500	000123		WRDATA		;WRDATA
5026	013502	177400		-400		;400 WORDS
5027	013504	063414		OBUFF		;OBUFF IS BUFF ADDRESS
5028	013506	021		.BYTE 21		;SECTOR 21
5029	013507	000		.BYTE 0		;TRACK 0
5030	013510	000312		312		;CYLINDER 312
5031	013512	004437	041740	JSR R4,GENCOM		;GENERATE DATA
5032	013516	000012		12		;PATTERN 12
5033	013520	000400		400		;400 WORDS
5034						
5035	013522	104417		TLOADRK		;LOAD RK REGS
5036	013524	104430		TWAT96		;WAIT FOR INTERRUPT
5037	013526	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
5038						
5039	013530	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5040	013532	104004		ERROR 4 ;OR 5, 6, 7,		;REPORT ALL ERRORS
5041	013534	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
5042	013540	000521		RDDATA:BA16		;READ DATA AND SET BA16
5043	013542	177400		-400		;400 WORDS
5044	013544	177770		177770		;ACROSS 64K BOUNDARY
5045	013546	021		.BYTE 21		;FROM SECTOR 21
5046	013547	000		.BYTE 0		;TRACK 0
5047	013550	000312		312		;CYLINDER 312
5048						
5049	013552	104417		TLOADRK		;LOAD RK REGS
5050	013554	104424		TWAT32		;WAIT FOR INTERRUPT
5051	013556	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
5052						
5053	013560	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5054	013562	104004		ERROR 4 ;OR 5, 6, 7,		;REPORT ALL ERRORS
5055	013564	004437	041740	JSR R4,GENCOM		;COMPARE DATA
5056	013570	120000		120000		;MEM MANAGEMENT WITH SOURCE
5057	013572	003777		3777		;RELOCATION ARGUMENT
5058	013574	000400		400		;400 WORDS
5059	013576	000413		BR 55		;NO MISCOMPARES-SKIP
5060	013600	104015		ERROR 15		;REPORT FIRST ERROR
5061						
5062	013602	013700	001634	MOV ERRMT,RO		;GET ERROR LIMIT
5063	013606	005300		DEC RO		;DECREMENT COUNT
5064	013610	001406		BEQ 655		;IF ZERO - EXIT
5065	013612	004437	041740	JSR R4,GENCOM		;CONTINUE DATA COMPARE
5066	013616	060000		60000		
5067	013620	000402		BR 655		;NO MORE ERRORS - EXIT
5068	013622	104016		ERROR 16		;REPORT NEXT ERROR
5069	013624	000770		BR 645		;LOOP
5070	013626					
5071						
5072	013626			55:		
5073				*****		
5074				TEST 36 WRITE DATA ADDRESS GREATER THAN 96K		
5075				*		
5076				* ISSUE A WRITE DATA OF 400 WORDS WITH ADDRESS = 577770.		
5077				* MAKE SURE CORRECT DATA IS ON DISK.		

H08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 97
T36 WRITE DATA ADDRESS GREATER THAN 96K

SEQ 0097

```

5078
5079
5080
5081
5082
5083 013626 000004
5084 013630 012737 000012 001262
5085 013636 123727 001326 000003
5086 013644 002016
5087 013646 032737 000002 001664 1S:
5088 013654 001011
5089
5090 013656 104401 051471
5091 013662 104401 051644
5092 013666 104401 051650
5093 013672 052737 000002 001664
5094 013700 000463 2S:
5095
5096 013702 012737 013710 001110 3S:
5097
5098 013710 4S:
5099 013710 104416
5100 013712 104003
5101 013714 004437 035530
5102 013720 001123
5103 013722 177400
5104 013724 177770
5105 013726 022
5106 013727 000
5107 013730 000312
5108 013732 004437 041740
5109 013736 010013
5110 013740 005777
5111 013742 000400
5112
5113 013744 104417
5114 013746 104430
5115 013750 104002
5116
5117 013752 104421
5118 013754 104004
5119
5120 013756 004437 035530
5121 013762 000121
5122 013764 177400
5123 013766 061414
5124 013770 022
5125 013771 000
5126 013772 000312
5127
5128 013774 104417
5129 013776 104424
5130 014000 104002
5131
5132 014002 104421
5133 014004 104004

```

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

T36: SCOPE ; DO 10. ITERATIONS
MOV #10, STIMES ; CHECK IF >96K MEMORY
CMPB \$MAMS1, #3 ; YES-SKIP
BGE 3S ; TEST IF REPORT FLAG SET
BIT #MEMSZB, OPTFLG ; NO-SKIP
BNE 2S

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

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

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

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

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

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

TLOADRK ; LOAD RK REGS
TWAT32 ; WAIT FOR INTERRUPT
ERROR 2 ; TO SLOW/NOT COMPLETE ERROR

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

```

5134
5135 014006 004437 041740      JSR      R4,GENCOM      ;COMPARE DATA
5136 014012 110000              110000      ;MEM MANAGEMENT WITH DESTINATION
5137 014014 005777              5777       ;RELOCATION ARGUMENT
5138 014016 000400              400        ;400 WORDS
5139 014020 000413              BR        6$      ;NO MISCOMPARES-BRANCH
5140 014022 104015              ERROR     15      ;REPORT 1ST ERROR
5141
5142 014024 013700 001634      MOV      ERRLMT,R0      ;GET ERROR LIMIT
5143 014030 005300      64$: DEC      R0      ;DECREMENT COUNT
5144 014032 001406              BEQ      65$      ;IF ZERO - EXIT
5145 014034 004437 041740      JSR      R4,GENCOM      ;CONTINUE DATA COMPARE
5146 014040 050000              50000
5147 014042 000402              BR        65$      ;NO MORE ERRORS - EXIT
5148 014044 104016              ERROR     16      ;REPORT NEXT ERROR
5149 014046 000770              BR        64$      ;LOOP
5150 014050      65$:
5151
5152 014050      6$:
5153 *****
5154 ;TEST 37      READ DATA ADDRESS GREATER THAN 96K
5155 ;
5156 ;      ISSUE A READ DATA OF 400 WORDS WITH ADDRESS = 577770.
5157 ;      CHECK MEMORY FOR CORRECT TRANSFER.
5158 ;
5159 ;      NOTE:  THIS TEST IS ONLY EXECUTED IF MORE THAN 96K
5160 ;      OF MEMORY IS PRESENT.
5161 ;
5162 *****
5163 014050 000004      ST37: SCOPE
5164 014052 012737 000012 001262      MOV      #10, $TIMES      ;DO 10. ITERATIONS
5165 014060 123727 001326 000003      CMPB     $MAMS1, #3      ;CHECK IF >96K MEMORY
5166 014066 002001      BGE      3$      ;YES-SKIP
5167 014070 000462      BR        6$
5168
5169 014072 012737 014100 001110      2$: BR
5170
5171 014100      3$: MOV      #4$, $LPERR      ;SET LOCAL LOOP ON ERROR
5172
5173 014100 104416      4$: TSSINIT
5174 014102 104003      ERROR     3      ;CLEAR SUBSYSTEM
5175
5176 014104 004437 035530      JSR      R4,LRLOAD      ;LOAD "L" REGS
5177 014110 000123      WRDATA      ;WRDATA
5178 014112 177400      -400      ;-400 WORDS
5179 014114 063414      OBUFF      ;OBUFF IS BUFF ADDRESS
5180 014116 005        .BYTE     5      ;SECTOR 5
5181 014117 000        .BYTE     0      ;TRACK 0
5182 014120 000312      312      ;CYLINDER 312
5183 014122 004437 041740      JSR      R4,GENCOM      ;GENERATE DATA
5184 014126 000014      14      ;PATTERN 14
5185 014130 000400      400      ;400 WORDS
5186
5187 014132 104417      TLOADRK      ;LOAD RK REGS
5188 014134 104430      TWAIT96      ;WAIT FOR INTERRUPT
5189 014136 104002      ERROR     2      ;TO SLOW/NOT COMPLETE ERROR

```

J08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 99
T37 READ DATA ADDRESS GREATER THAN 96K

SEQ 0099

5190	014140	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5191	014142	104004		ERROR 4	;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
5192	014144	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
5193	014150	001121		RDDATA:BA17		;READ DATA WITH BA17 SET
5194	014152	177400		-400		;400 WORDS
5195	014154	177770		177770		;ACROSS 96K BOUNDARY
5196	014156	005		.BYTE 5		;FROM SECTOR 5
5197	014157	000		.BYTE 0		;TRACK 0
5198	014160	000312		312		;CYL 312
5199						
5200	014162	104417		TLOADRK		;LOAD RK REGS
5201	014164	104424		TWAT32		;WAIT FOR INTERRUPT
5202	014166	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
5203						
5204	014170	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5205	014172	104004		ERROR 4	;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
5206	014174	004437	041740	JSR R4,GENCOM		;COMPARE DATA
5207	014200	120000		120000		;MEM MANAGEMENT WITH SOURCE
5208	014202	005777		5777		;RELOCATION ARGUMENT
5209	014204	000400		400		;400 WORDS
5210	014206	000413		BR 65		;NO MISCOMPARES-SKIP
5211	014210	104015		ERROR 15		;REPORT FIRST ERROR
5212						
5213	014212	013700	001634	MOV ERLMT,R0		;GET ERROR LIMIT
5214	014216	005300		DEC R0		;DECREMENT COUNT
5215	014220	001406		BEO 655		;IF ZERO - EXIT
5216	014222	004437	041740	JSR R4,GENCOM		;CONTINUE DATA COMPARE
5217	014226	060000		60000		
5218	014230	000402		BR 655		;NO MORE ERRORS - EXIT
5219	014232	104016		ERROR 16		;REPORT NEXT ERROR
5220	014234	000770		BR 645		;LOOP
5221	014236					
5222						
5223	014236			65:		
5224				*****		
5225				;TEST 40 PARTIAL SECTOR WRITE DATA		
5226				*****		
5227				ISSUE A WRITE DATA OF 103 WORDS TO CYLINDER 312,		
5228				HEAD 0 SECTOR 0. MAKE SURE THE SECTOR WAS		
5229				ZERO FILLED CORRECTLY.		
5230				*****		
5231				*****		
5232	014236	000004		TEST40: SCOPE		
5233	014240	012737	000012 001262	MOV #10.,\$TIMES		;DO 10. ITERATIONS
5234	014246	104416		TSSINIT		;CLEAR SUBSYSTEM
5235	014250	104003		ERROR 3		;BAD INIT ERROR
5236						
5237	014252	004437	035530	JSR R4,LRLOAD		;LOAD "L" REG
5238	014256	000123		WRDATA		;WRITE DATA
5239	014260	177675		-103		;WORD COUNT PARTIAL SECTOR
5240	014262	063414		OBUFF		;BUFF ADDRESS
5241	014264	007		.BYTE 7		;SECTOR 7
5242	014265	000		.BYTE 0		;TRACK 0
5243	014266	000312		312		;CYLINDER 312
5244						
5245	014270	004437	041740	JSR R4,GENCOM		;GENERATE DATA

K08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 100
DZR6KD.P11 31-JAN-77 16:26 T40 PARTIAL SECTOR WRITE DATA

SEQ 0100

```

5246 014274 000003          3          ;PATTERN 3
5247 014276 000400          400        ;400 WORDS
5248
5249 014300 104417          TLOADRK      ;LOAD RK REGS
5250 014302 104430          TWAT96      ;WAIT FOR INTERRUPT
5251 014304 104002          ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
5252
5253 014306 104421          TCHKOP      ;CHECK FOR ANY ERROR
5254 014310 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERROR
5255
5256 014312 004437 035530    JSR      R4,LRLOAD ;LOAD "L" REGS
5257
5258 014316 000121          RDDATA      ;READ DATA
5259 014320 177400          -400        ;ONE FULL SECTOR
5260 014322 061414          Ibuff      ;BUFF ADDRESS
5261 014324 007           .BYTE 7      ;SECTOR 7
5262 014325 000           .BYTE 0      ;TRACK 0
5263 014326 000312          312        ;CYLINDER 312
5264
5265 014330 004437 041740    JSR      R4,GENCOM
5266 014334 002007          2007        ;CLEAR Ibuff TO ALL ONES
5267 014336 000400          400
5268
5269 014340 104417          TLOADRK      ;LOAD RK REGS
5270 014342 104424          TWAT32      ;WAIT FOR INTERRUPT
5271 014344 104002          ERROR 2      ;TO SLOW/NOT COMPLETE
5272
5273 014346 104421          TCHKOP      ;CHECK FOR ANY ERRORS
5274 014350 104004          ERROR 4 ; OR 5,6,7 ;REPORT ALL ERRORS
5275
5276 014352 012701 063622    MOV      #OBUFF+206,R1 ;CLEAR THE LAST 205 WORDS
5277 014356 012700 000275    MOV      #275,R0 ;OF THE OUTPUT BUFFER TO ZERO
5278 014362 005021          CLR      (R1)+ ;TO VERIFY THE PARTIAL SECTOR
5279 014364 005300          DEC      R0 ;WRITE 0 FILLED THE SECTOR
5280 014366 001375          BNE      1$
5281 014370 004437 041740    JSR      R4,GENCOM
5282 014374 100000          100000      ;COMPARE OBUFF & Ibuff.
5283 014376 000400          400        ;ALL 400 WORDS
5284 014400 000413          BR      3$   ;NO ERRORS-EXIT
5285 014402 104015          ERROR 15    ;REPORT FIRST COMPARE ERROR
5286
5287 014404 013700 001634    MOV      ERRLMT,R0 ;GET ERROR LIMIT
5288 014410 005300          DEC      R0 ;DECREMENT IT
5289 014412 001406          BEQ      3$   ;IF ZERO-EXIT
5290 014414 004437 041740    JSR      R4,GENCOM
5291 014420 040000          40000      ;CONTINUE COMPARE
5292 014422 000402          BR      3$   ;NO MORE ERRORS-EXIT
5293 014424 104016          ERROR 16    ;REPORT NEXT COMPARE ERROR
5294 014426 000770          BR      2$   ;LOOP
5295
5296 014430          3$:
5297 *****
5298 *TEST 41 PARTIAL SECTOR READ DATA
5299 *
5300 * WRITE CYLINDER 312, TRACK 0, SECTOR ZERO WITH A
5301 * KNOWN CONFIGURATION. ISSUE A READ DATA OF

```

L08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 101
T41 PARTIAL SECTOR READ DATA

SEQ 0101

```

5302      *      103 WORDS TO CYLINDER 312, TRACK 0, SECTOR 0.
5303      *      MAKE SURE ONLY 103 WORDS GET TRANSFERRED
5304      *      TO MEMORY.
5305      *
5306      * *****
5307 014430 000004      ST41: SCOPE
5308 014432 012737 000012 001262      MOV      #10., $TIMES      ; DO 10. ITERATIONS
5309 014440 104416      TSSINIT      ; CLEAR SUBSYSTEM
5310 014442 104003      ERROR      3      ; BAD INIT ERROR
5311
5312 014444 004437 035530      JSR      R4, LRLOAD      ; LOAD "L" REGS
5313 014450 000123      WDATA      ; WDATA
5314 014452 177400      -400      ; -400 WORDS
5315 014454 063414      OBUF      ; OBUF IS BUFF ADDRESS
5316 014456 017      .BYTE      17      ; SECTOR 17
5317 014457 000      .BYTE      0      ; TRACK 0
5318 014460 000312      312      ; CYLINDER 312
5319 014462 004437 041740      JSR      R4, GENCOM      ; GENERATE DATA
5320 014466 000004      4      ; PATTERN 4
5321 014470 000400      400      ; 400 WORDS
5322
5323 014472 104417      TLOADRK      ; LOAD RK REGS
5324 014474 104430      TWAT96      ; WAIT FOR INTERRUPT
5325 014476 104002      ERROR      2      ; TO SLOW/NOT COMPLETE ERROR
5326
5327 014500 104421      TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
5328 014502 104004      ERROR      4 ;OR 5, 6, 7, 10 ; REPORT ALL ERRORS
5329
5330 014504 004437 035530      JSR      R4, LRLOAD      ; LOAD "L" REGS
5331 014510 000121      RDATA      ; RDATA
5332 014512 177675      -103      ; -103 WORDS
5333 014514 061414      IBUF      ; IBUF IS BUFF ADDRESS
5334 014516 017      .BYTE      17      ; SECTOR 17
5335 014517 000      .BYTE      0      ; TRACK 0
5336 014520 000312      312      ; CYLINDER 312
5337 014522 004437 041740      JSR      R4, GENCOM      ; GENERATE DATA
5338 014526 002007      2007      ; CLEAR IBUF
5339 014530 000400      400
5340
5341 014532 104417      TLOADRK      ; LOAD RK REGS
5342 014534 104424      TWAT32      ; WAIT FOR INTERRUPT
5343 014536 104002      ERROR      2      ; TO SLOW/NOT COMPLETE ERROR
5344 014540 104421      TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
5345 014542 104004      ERROR      4 ;OR 5, 6, 7, 10 ; REPORT ALL ERRORS
5346
5347 014544 012700 063622      MOV      #OBUF+206, R0      ; AFTER THE LAST 205 WORDS OF
5348 014550 012701 000275      MOV      #275, R1      ; THE OUTPUT BUFFER TO ALL ONES.
5349 014554 012720 177777      MOV      #-1, (R0)+      ; THESE SHOULD ALL BE ONES IN
5350 014560 005301      DEC      R1      ; IBUF BECAUSE THE PARTIAL
5351 014562 001374      BNE      1$      ; READ FILLED ONLY 103 WORDS.
5352 014564 004437 041740      JSR      R4, GENCOM      ; GO COMPARE IBUF & OBUF
5353 014570 100000      100000
5354 014572 000400      400      ; ALL 400 WORDS
5355 014574 000413      BR      3$      ; NO ERRORS-EXIT
5356 014576 104015      ERROR      15      ; REPORT FIRST COMPARE ERROR
5357

```

M08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 102
T41 PARTIAL SECTOR READ DATA

SEQ 0102

5358	014600	013700	001634		MOV	ERRLMT, R0	; GET ERROR LIMIT
5359	014604	005300			DEC	R0	; DECREMENT COUNT
5360	014606	001406			BEQ	65\$	; IF ZERO - EXIT
5361	014610	004437	041740		JSR	R4, GENCOM	; CONTINUE DATA COMPARE
5362	014614	040000				40000	
5363	014616	000402			BR	65\$	; NO MORE ERRORS - EXIT
5364	014620	104016			ERROR	16	; REPORT NEXT ERROR
5365	014622	000770			BR	64\$	; LOOP
5366	014624						
5367							
5368	014624						
5369							
5370							
5371							
5372							
5373							
5374							
5375							
5376	014624	000004					
5377	014626	012737	000012	001262			
5378	014634	104416					
5379	014636	104003					
5380							
5381	014640	004437	035530		JSR	R4, LRLOAD	; LOAD "L" REG
5382	014644	001523			BA16!BA17!WRDATA		; BA16 & 17 SET WITH WRITE DATA
5383	014646	177777			-1		; WORD COUNT OF 1
5384	014650	176000			176000		; BUFF ADDRESS=10 PAGE BASE
5385	014652	013			.BYTE	13	; SECT 13
5386	014653	000			.BYTE	0	; TRACK 0
5387	014654	000312			312		; CYLINDER 312
5388							
5389	014656	104417			TLOADRK		; LOAD RK REGS
5390	014660	104430			TWAIT96		; WAIT FOR INTERRUPT
5391	014662	104002			ERROR	2	; TO SLOW/NOT COMPLETE ERROR
5392	014664	104422			TCHKWE		; CHECK OPERATION WITH ERROR
5393	014666	000000			0		
5394	014670	000000			0		
5395	014672	000040			NEMERR		; NON-EXISTENT MEMORY ERROR
5396	014674	104004			ERROR	4	; REPORT ANY DISCREPENCIES
5397	014676	012737	054047	001450	MOV	#EM11A, EM10N	; SET UP ERROR MESSAGE
5398	014704	012737	050240	060770	MOV	#OPER42, OF010A	; WITH SUPPORT MESSAGE
5399	014712	113700	001541		MOV	T.CS1+1, R0	; GET UPPER CS1
5400	014716	042700	177774		BIC	#177774, R0	; CLEAR UNUSED BITS
5401	014722	022700	000003		CMP	#3, R0	; TEST IF BOTH UPPER BUS BITS SET
5402	014726	001406			BEQ	1\$	; YES - SKIP
5403	014730	010037	001204		MOV	R0, \$REG11	; SET UP FOR ERROR REPORT
5404	014734	012737	000003	001202	MOV	#3, \$REG10	
5405	014742	104010			ERROR	10	
5406	014744	022737	176002	001544	1\$: CMP	#176002, T.BA	; TEST IF BUSS ADDRESS LOW OKAY
5407	014752	001412			BEQ	2\$	; YES - SKIP
5408	014754	012737	054021	001450	MOV	#EM11, EM10N	; SET UP MESSAGE
5409	014762	012737	176002	001202	MOV	#176002, \$REG10	; STORE VALUE FOR REPORT
5410	014770	013737	001544	001204	MOV	T.BA, \$REG11	
5411	014776	104010			ERROR	10	
5412							
5413	015000	005737	001542		2\$: TST	T.WC	; TEST IF WORD COUNT CORRECT

N08

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 103
T42 WRITE DATA WITH NON-EXISTENT MEMORY

SEQ 0103

5414 015004 001411
5415 015006 012737 053774 001450
5416 015014 005037 001202
5417 015020 013737 001542 001204
5418 015026 104010
5419
5420 015030
5421
5422
5423
5424
5425
5426
5427
5428
5429 015030 000004
5430 015032 012737 000012 001262
5431 015040 104416
5432 015042 104003
5433
5434 015044 004437 035530
5435 015050 001521
5436 015052 177777
5437 015054 176000
5438 015056 013
5439 015057 000
5440 015060 000312
5441
5442 015062 104417
5443 015064 104430
5444 015066 104002
5445 015070 104422
5446 015072 000000
5447 015074 000000
5448 015076 000040
5449 015100 104004
5450 015102 012737 054047 001450
5451 015110 012737 050314 060770
5452 015116 113700 001541
5453 015122 042700 177774
5454 015126 022700 000003
5455 015132 001406
5456 015134 012737 000003 001202
5457 015142 010037 001204
5458 015146 104010
5459
5460 015150 022737 176002 001544 15:
5461 015156 001412
5462 015160 012737 054021 001450
5463 015166 012737 176002 001202
5464 015174 013737 001544 001204
5465 015202 104010
5466
5467 015204 005737 001542 25:
5468 015210 001411
5469 015212 012737 053774 001450

BEQ 3\$;YES - SKIP
MOV #EM10,EM10N ;SET UP MESSAGE
CLR \$REG10
MOV T.WC,\$REG11
ERROR 10

3\$:

TEST 43 READ DATA WITH NON-EXISTENT MEMORY

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

↑ST43: SCOPE
MOV #10.,\$TIMES ;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REG
BA16!BA17!RDATA ;BA16 & 17 WITH READ DATA
-1 ;WORD COUNT OF 1
176000 ;BUFF ADDRESS=10 PAGE BASE
.BYTE 13 ;SECTOR 13
.BYTE 0 ;TRACK 0
312 ;CYL 312
TLOADRK ;LOAD RK REGS
TWAT96 ;WAIT FOR INTERRUPT
ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
TCHKWE ;CHECK OPERATION WITH ERRORS
0
0
NEMERR ;NON-EXISTENT MEMORY ERROR
ERROR 4: OR 5,6,7 ;REPORT ALL DISCREPANCIES
MOV #EM11A,EM10N ;SET MESSAGE
MOV #OPER43,DF010A ;SET SUPPORT MESSAGE
MOVB T.CS1+1,R0 ;GET UPPER CS1
BIC #177774,R0 ;CLEAR UNWANTED BITS
CMP #3,R0 ;TEST BOTH BUS 16 & 17 SET
BEQ 1\$;YES - SKIP
MOV #3,\$REG10 ;SET VALUES FOR REPORT
MOV R0,\$REG11
ERROR 10

CMP #176002,T.BA ;TEST IF BUS ADDRESS CORRECT
BEQ 2\$;YES - SKIP
MOV #EM11,EM10N ;SET MESSAGE
MOV #176002,\$REG10 ;SET VALUES FOR REPORT
MOV T.BA,\$REG11
ERROR 10

TST T.WC ;TEST IF WORD COUNT CORRECT
BEQ 3\$;YES - SKIP
MOV #EM10,EM10N ;SET MESSAGE

5470	015220	005037	001202		CLR	\$REG10		;SET VALUES
5471	015224	013737	001542	001204	MOV	T.WC,\$REG11		
5472	015232	104010			ERROR	10		
5473								
5474	015234				3\$:			
5475								
5476								
5477								
5478								
5479								
5480								
5481								
5482								
5483								
5484								
5485								
5486								
5487								
5488								
5489								
5490	015234	000004						
5491	015236	012737	000012	001262	ST44:	SCOPE		
5492	015244	104416				MOV	#10.,\$TIMES	;DO 10. ITERATIONS
5493	015246	104003				TSSINIT		;CLEAR SUBSYSTEM
5494						ERROR	3	;BAD INIT ERROR
5495	015250	032737	000100	001664		BIT	#PARPRE,OPTFLG	;TEST IF PARITY OPTION PRESENT
5496	015256	001013				BNE	1\$	;YES-SKIP
5497	015260	032737	000004	001664		BIT	#MEMPYB,OPTFLG	;TEST IF PARITY OPTION REPORTED
5498	015266	001005				BNE	25\$	;YES-SKIP TO EXIT
5499	015270	104401	051274			TYPE	OPR010	;PRINT BYPASS MESSAGE
5500	015274	052737	000004	001664		BIS	#MEMPYB,OPTFLG	;SET OPTION REPORTED BIT
5501	015302	000137	016050		25\$:	JMP	2\$	;SKIP TO EXIT
5502								
5503	015306				1\$:			
5504	015306	004437	035530			JSR	R4,LRLOAD	;LOAD "L" REGS
5505	015312	000123				WRDATA		;WRDATA
5506	015314	177400				-400		; -400 WORDS
5507	015316	063414				OBUFF		;OBUFF IS BUFF ADDRESS
5508	015320	010				.BYTE	10	;SECTOR 10
5509	015321	000				.BYTE	0	;TRACK 0
5510	015322	0 312				312		;CYLINDER 312
5511								
5512	015324	032737	002000	001664		BIT	#CP1170,OPTFLG	;TEST IF 11/70
5513	015332	001002				BNE	3\$	;YES - SKIP
5514	015334	005077	164332			CLR	2CSRPT	;CLEAR PARITY IE
5515								
5516	015340	004437	041740		3\$:	JSR	R4,GENCOM	;GENERATE DATA
5517	015344	000007				7		;PATTERN 7
5518	015346	000400				400		;400 WORDS
5519								
5520	015350	012746	000340			MOV	#PR7,-(SP)	;PUT PRIORITY 7 ON STACK
5521	015354	012746	015362			MOV	#10\$,-(SP)	;PUT ADDRESS ON STACK
5522	015360	000002				RTI		;SET PRI
5523	015362				10\$:			
5524								
5525	015362	012737	100200	063640		MOV	#100200,OBUFF+224	;SET WORD IN BUFFER

C09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 105
T44 UNIBUS PARITY ERROR

SEQ 0105

```

5526 015370 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5527 015376 001011                    BNE      55$          ;YES - SKIP
5528 015400 012777 000004 164264      MOV      #BIT2,2CSRPTR ;SET WRITE WRONG PARITY BIT
5529 015406 012737 100200 063640      MOV      #100200,0BUFF+224 ;SET BAD PARITY IN MEMORY
5530 015414 012777 000001 164250      MOV      #BIT0,2CSRPTR ;CLEAR CONTROL BIT, SET IE BIT
5531
5532 015422 013746 001622                55$: MOV      RKPRI,-(SP) ;SET OLD PRIORITY
5533 015426 012746 015434                MOV      #11$,-(SP) ;SET ADDRESS
5534 015432 000002                    RTI      ;RESTORE PRI
5535 015434 013704 001672                11$: MOV      CSRPTR,R4 ;SET R4 WITH CSR POINTER
5536 015440 005000                    CLR      R0 ;SET R0 FOR COUNTER
5537 015442 012701 001662                MOV      #INTSET,R1 ;SET R1 FOR POINTER TO INTERRUPT FLAG
5538 015446 012777 061400 164144      MOV      #SPCHLR,2RKVEC ;SET UP INTERRUPT VECTORS FOR
5539 015454 012777 061410 164234      MOV      #SPCPAR,2RMVECA ;RK611 AND PARITY ERROR
5540 015462 012737 016050 001264      MOV      #2$,2$ESCAPE ;SET UP ESCAPE FOR ERROR
5541 015470 104417                    TLOADRK ;LOAD RK REGS
5542 015472 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5543 015510 001434                    BEQ      45$          ;NO - SKIP
5544 015512 012737 000016 177746      MOV      #16,177746 ;SET TO DISABLE CACHE AND UNIBUS ERROR
5545 015513 012777 000000 164172      MOV      #0,2$WLOAD ;TURN OFF CLOCK INTERRUPTS
5546 015516 012777 000000 163420      MOV      #0,2$TKS ;TURN OFF KEYBOARD
5547 015524 012737 170000 177750      MOV      #170000,177750 ;SET FOR ERROR FORCE
5548
5549
5550 015532 105711                    ;*****
5551 015534 003005      40$: TSTB      (R1) ;LOOP TO WAIT FOR INTERRUPT OR ABORT
5552 015536 005300                    BGT      43$          ;WAIT. THE CODE BETWEEN THE STARS IS SET
5553 015540 100774                    DEC      R0 ;SET UP SO ALL BYTES HAVE PARITY OF 1.
5554 015542 000240                    BMI      40$          ;IF THIS CODE IS CHANGED, REMEMBER ALL
5555 015544 003372                    NOP ;BYTES MUST HAVE AN EVEN NUMBER OF
5556 015546 000240                    BGT      40$          ;BITS.
5557 015550 005014                    NOP
5558                                43$: CLR      (R4) ;CLEAR ERROR FORCE
5559                                ;*****
5560 015552 005037 177746                    CLR      177746 ;ENABLE CACHE
5561 015556 012777 000100 164124      MOV      #BIT6,2$WLOAD ;TURN ON CLOCK INTERRUPTS
5562 015564 012777 000100 163352      MOV      #100,2$TKS ;TURN ON KEYBOARD INTERRUPTS
5563 015572 104430                    45$: TWAT96 ;WAIT FOR INTERRUPT
5564 015574 000414                    BR       46$          ;TO SLOW/NOT COMPLETE ERROR
5565 015576 032737 002000 001664      BIT      #CP1170,OPTFLG ;TEST IF 11/70
5566 015604 001024                    BNE      48$          ;YES - SKIP
5567 015606 005077 164060                    CLR      2CSRPTR ;ELSE CLEAR CSR
5568 015612 005037 063640                    CLR      0BUFF+224 ;CLEAR THE BAD PARITY WORD
5569 015616 012777 000001 164046      MOV      #1,2CSRPTR ;SET PARITY DETECT AGAIN
5570 015624 000414                    BR       48$          ;SKIP
5571
5572 015626 032737 002000 001664      46$: BIT      #CP1170,OPTFLG ;TEST IF 11/70
5573 015634 001007                    BNE      47$          ;YES - SKIP
5574 015636 005077 164030                    CLR      2CSRPTR ;CLEAR CSR
5575 015642 005037 063640                    CLR      0BUFF+224 ;CLEAR BAD PARITY WORD
5576 015646 012777 000001 164016      MOV      #1,2CSRPTR ;SET UP PARITY DETECT AGAIN
5577 015654 104002                    47$: ERROR 2 ;REPORT TO SLOW ERROR
5578 015656                    48$:
5579 015656 104422                    TCHKWE ;CHECK OPERATION WITH ERROR
5580 015660 000000                    0
5581 015662 000000                    0

```

```

5582 015664 000100          UPERR          ;UNIBUS PARITY ERROR
5583 015666 104004          ERROR 4; OR 5,6,7 ;REPORT ALL DISCREPANCIES
5584
5585 015670 005037 001264    CLR          $ESCAPE          ;CLEAR ESCAPE
5586 015674 012737 050367 060770 MOV      $OPERR4,DF010A ;SET MESSAGES FOR REPORT
5587 015702 012737 054021 001450 MOV      $EM11,EM10N
5588 015710 023727 001544 063642 CMP      T.BA,$OBUFF+226 ;CHECK IF BA IN RANGE
5589 015716 103010          BHS          14$          ;NOT TO LOW - SKIP
5590 015720 012737 063642 001202 MOV      $OBUFF+226,$REG10 ;SET VALUES FOR REPORT
5591 015726 013737 001544 001204 MOV      T.BA,$REG11
5592 015734 104010          ERROR          10
5593 015736 000413          BR          16$
5594
5595 015740 023727 001544 063646 14$: CMP      T.BA,$OBUFF+232 ;CHECK IF BA IN RANGE
5596 015746 101407          BLOS         16$          ;YES - SKIP
5597 015750 012737 063646 001202 MOV      $OBUFF+232,$REG10 ;SET VALUES
5598 015756 012737 001544 001204 MOV      T.BA,$REG11
5599 015764 104010          ERROR          10
5600
5601 015766 012737 053774 001450 16$: MOV      $EM10,EM10N          ;SET MESSAGE
5602 015774 023727 001542 177513 CMP      T.WC,$-265          ;CHECK IF WORD COUNT WITHIN RANGE
5603 016002 103007          BHS          20$          ;YES - SKIP
5604 016004 012737 177513 001202 MOV      $-265,$REG10          ;SET VALUES
5605 016012 013737 001542 001204 MOV      T.WC,$REG11
5606 016020 104010          ERROR          10
5607
5608 016022 023727 001542 177515 20$: CMP      T.WC,$-263          ;STILL CHECKING IF WC IN RANGE
5609 016030 101407          BLOS         2$          ;YES - SK ?
5610 016032 012737 177515 001202 MOV      $-263,$REG10          ;SET VALUES
5611 016040 013737 001542 001204 MOV      T.WC,$REG11
5612 016046 104010          ER          10
5613
5614 016050 012777 034354 163542 2$: MOV      $INTHLR,$RKVEC          ;RESET INT. VECTORS FOR RK06
5615 016056 012777 034364 163632 MOV      $PERHLR,$MMVECA          ;AND PARITY ERRORS
5616 016064 012777 000100 163616 MOV      $BIT6,$KLADD          ;INSURE CLOCK IS ENABLED
5617 016072 012777 000100 163044 MOV      $100,$STKS          ;INSURE KEYBOARD IS ENABLED
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630 016100 000004          .SBTTL  **MULTIPLE SECTOR OPERATIONS
5631 016102 012737 000012 001262 ;*****
5632 016110 104416          ;TEST 45          TWO SECTOR WRITE DATA (PART 1)
5633 016112 104003          ;
5634
5635 016114 004437 035530          ;ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5636 016120 000123          ;TRACK 0, SECTOR 0. READ DATA BACK ONE SECTOR
5637 016122 177000          ;AT A TIME AND MAKE SURE IT IS CORRECT.
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

```

E09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 107
 DZR6KD.P11 31-JAN-77 16:26 T45 TWO SECTOR WRITE DATA (PART 1)

SEQ 0107

5638	016124	063414		OBUFF		;OBUFF IS BUFF ADDRESS
5639	016126	000		.BYTE	0	;SECTOR 0
5640	016127	000		.BYTE	0	;TRACK 0
5641	016130	000312		312		;CYLINDER 312
5642						
5643	016132	004437	041740	JSR	R4,GENCOM	;GENERATE DATA
5644	016136	000015		15		;PATTERN 15
5645	016140	001000		1000		;1000 WORDS
5646						
5647	016142	104417		TLOADRK		;LOAD RK REGS
5648	016144	104430		TWAT96		;WAIT FOR INTERRUPT
5649	016146	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5650						
5651	016150	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5652	016152	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5653						
5654	016154	004437	041740	JSR	R4,GENCOM	;CLEAR Ibuff
5655	016160	002007		2007		;TO ALL 1'S
5656	016162	001000		1000		
5657						
5658	016164	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
5659	016170	000121		RDDATA		;RDDATA
5660	016172	177400		-400		; -400 WORDS
5661	016174	061414		IBUFF		;IBUFF IS BUFF ADDRESS
5662	016176	000		.BYTE	0	;SECTOR 0
5663	016177	000		.BYTE	0	;TRACK 0
5664	016200	000312		312		;CYLINDER 312
5665						
5666	016202	104417		TLOADRK		;LOAD RK REGS
5667	016204	104424		TWAT32		;WAIT FOR INTERRUPT
5668	016206	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5669						
5670	016210	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5671	016212	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5672						
5673	016214	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
5674	016220	000121		RDDATA		;RDDATA
5675	016222	177400		-400		; -400 WORDS
5676	016224	062414		IBUFF+1000		;IBUFF+1000 IS BUFF ADDRESS
5677	016226	001		.BYTE	1	;SECTOR 1
5678	016227	000		.BYTE	0	;TRACK 0
5679	016230	000312		312		;CYLINDER 312
5680						
5681	016232	104417		TLOADRK		;LOAD RK REGS
5682	016234	104424		TWAT32		;WAIT FOR INTERRUPT
5683	016236	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5684						
5685	016240	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5686	016242	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5687						
5688	016244	004437	041740	JSR	R4,GENCOM	;COMPARE DATA
5689	016250	100000		100000		
5690	016252	001000		1000		;1000 WORDS
5691	016254	000413		BR	25	;NO MISCOMPARES-EXIT
5692	016256	104015		ERROR	15	;REPORT FIRST ERROR
5693						

F09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 108
T45 TWO SECTOR WRITE DATA (PART 1)

SEQ 0108

5694 016260 013700 001634
 5695 016264 005300
 5696 016266 001406
 5697 016270 004437 041740
 5698 016274 040000
 5699 016276 000402
 5700 016300 104016
 5701 016302 000770
 5702 016304
 5703
 5704 016304
 5705
 5706
 5707
 5708
 5709
 5710
 5711
 5712
 5713

5714 016304 000004
 5715 016306 012737 000012 001262
 5716 016314 104416
 5717 016316 104003
 5718
 5719 016320 004437 035530
 5720 016324 000123
 5721 016326 177000
 5722 016330 063414
 5723 016332 023
 5724 016333 000
 5725 016334 000312
 5726
 5727 016336 004437 041740
 5728 016342 000016
 5729 016344 001000
 5730
 5731 016346 104417
 5732 016350 104430
 5733 016352 104002
 5734
 5735 016354 104421
 5736 016356 104004
 5737
 5738
 5739
 5740
 5741

5742 016360 004437 035530
 5743 016364 000121
 5744 016366 177400
 5745 016370 061414
 5746 016372 023
 5747 016373 000
 5748 016374 000312
 5749

645: MOV ERRMT,RO ;GET ERROR LIMIT
 DEC RO ;DECREMENT COUNT
 BEQ 655 ;IF ZERO - EXIT
 JSR R4,GENCOM ;CONTINUE DATA COMPARE
 40000
 BR 655 ;NO MORE ERRORS - EXIT
 ERROR 16 ;REPORT NEXT ERROR
 BR 645 ;LOOP

655:

25:

 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.
 *

TST46:

SCOPE
 MOV #10.,\$TIMES ;DO 10. ITERATIONS
 TSSINIT ;CLEAR SUBSYSTEM
 ERROR 3 ;BAD INIT ERROR
 JSR R4,LRLOAD ;LOAD "L" REGS
 WRDATA ;WRDATA
 -1000 ;-1000 WORDS
 OBUFF ;OBUFF IS BUFF ADDRESS
 .BYTE 23 ;SECTOR 23
 .BYTE 0 ;TRACK 0
 312 ;CYLINDER 312
 JSR R4,GENCOM ;GENERATE DATA
 16 ;PATTERN 16
 1000 ;1000 WORDS
 TLOADRK ;LOAD RK REGS
 TWAT96 ;WAIT FOR INTERRUPT
 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS

IF THE TRACK ADDRESS AT THE END OF OPERATION IS IN ERROR
 THE CONTROLLER DID A MID-TRANSFER SEEK AS THOUGH IT
 WERE IN 24(8) SECTORS PER TRACK MODE.

JSR R4,LRLOAD ;LOAD "L" REGS
 RDATA ;RDATA
 -400 ;-400 WORDS
 IBUFF ;IBUFF IS BUFF ADDRESS
 .BYTE 23 ;SECTOR 23
 .BYTE 0 ;TRACK 0
 312 ;CYLINDER 312

G09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR&KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 109
T46 TWO SECTOR WRITE DATA (PART 2)

SEQ 0109

```

5750 016376 004437 041740      JSR      R4,GENCOM      ;CLEAR Ibuff TO ALL ONES
5751 016402 002007              2007
5752 016404 001000              1000
5753
5754 016406 104417      TLOADRK      ;LOAD RK REGS
5755 016410 104424      TWAT32      ;WAIT FOR INTERRUPT
5756 016412 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
5757
5758 016414 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
5759 016416 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5760
5761 016420 004437 035530      JSR      R4,LRLOAD      ;LOAD "L" REGS
5762 016424 000121      RDATA      ;RDATA
5763 016426 177400      -400      ;-400 WORDS
5764 016430 062414      Ibuff+1000 ;IBuff+1000 IS BUFF ADDRESS
5765 016432 024        .BYTE 24    ;SECTOR 24
5766 016433 000        .BYTE 0     ;TRACK 0
5767 016434 000312      312        ;CYLINDER 312
5768
5769 016436 104417      TLOADRK      ;LOAD RK REGS
5770 016440 104424      TWAT32      ;WAIT FOR INTERRUPT
5771 016442 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
5772
5773 016444 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
5774 016446 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5775
5776 016450 004437 041740      JSR      R4,GENCOM      ;COMPARE DATA
5777 016454 100000      100000
5778 016456 001000      1000      ;1000 WORDS
5779 016460 000413      BR 15       ;NO ERRORS-SKIP
5780 016462 104015      ERROR 15    ;REPORT FIRST ERROR
5781
5782 016464 013700 001634      MOV      ERRLMT,R0      ;GET ERROR LIMIT
5783 016470 005300      64$: DEC R0      ;DECREMENT COUNT
5784 016472 001406      BEQ 65$      ;IF ZERO - EXIT
5785 016474 004437 041740      JSR      R4,GENCOM      ;CONTINUE DATA COMPARE
5786 016500 040000      40000
5787 016502 000402      BR 65$      ;NO MORE ERRORS - EXIT
5788 016504 104016      ERROR 16    ;REPORT NEXT ERROR
5789 016506 000770      BR 64$      ;LOOP
5790
5791 016510      65$:
5792 13:
5793 *****
5794 *TEST 47 TWO SECTOR WRITE DATA (PART 3)
5795 *
5796 * ISSUE A WRITE DATA OF 401 WORDS TO CYLINDER 312,
5797 * TRACK 0, SECTOR 10. READ DATA BACK ONE SECTOR AT
5798 * A TIME AND CHECK ZERO FILL OF SECOND SECTOR.
5799 *
5800 *****
5801 016510 000004      1ST47: SCOPE
5802 016512 012737 000012 001262      MOV #10.,$TIMES ;DO 10. ITERATIONS
5803 016520 104416      TSSINIT      ;CLEAR SUBSYSTEM
5804 016522 104003      ERROR 3      ;BAD INIT ERROR
5805 016524 004437 035530      JSR      R4,LRLOAD      ;LOAD "L" REGS

```

H09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 110
T47 TWO SECTOR WRITE DATA (PART 3)

SEQ 0110

5806	016530	000123		WRDATA	;WRDATA
5807	016532	177377		-401	; -401 WORDS
5808	016534	063414		OBUFF	;OBUFF IS BUFF ADDRESS
5809	016536	010		.BYTE 10	;SECTOR 10
5810	016537	000		.BYTE 0	;TRACK 0
5811	016540	000312		312	;CYLINDER 312
5812					
5813	016542	004437	041740	JSR R4,GENCOM	;GENERATE DATA
5814	016546	000002		2	;PATTERN 2
5815	016550	001000		1000	;1000 WORDS
5816					
5817	016552	104417		TLOADRK	;LOAD RK REGS
5818	016554	104430		TWAT96	;WAIT FOR INTERRUPT
5819	016556	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
5820					
5821	016560	104421		TCHKOP	;CHECK OPERATION FOR ANY ERRORS
5822	016562	104004		ERROR 4 ;OR 5 6 7	10 ;REPORT ALL ERRORS
5823				CLEAR LAST 377 WORDS OF	OBUFF FOR EXPECTED ZEROS FROM ZERO FILL
5824	016564	012700	064416	MOV #OBUFF+1002,R0	;GET STARTING ADDRESS TO BE CLEARED
5825	016570	012701	000377	MOV #377,R1	;NUMBER OF WORDS
5826	016574	005020		CLR (R0)+	;CLEAR WORD
5827	016576	005301		DEC R1	;DEC COUNT
5828	016600	001375		BNE 15	;LOOP UNTIL COUNT ZERO
5829	016602	004437	041740	JSR R4,GENCOM	;CLEAR IBUFF TO ONES
5830	016606	002007		2007	
5831	016610	001000		1000	
5832					
5833	016612	004437	035530	JSR R4,LRLoad	;LOAD "L" REGS
5834	016616	000121		RDDATA	;RDDATA
5835	016620	177400		-400	; -400 WORDS
5836	016622	061414		IBUFF	;IBUFF IS BUFF ADDRESS
5837	016624	010		.BYTE 10	;SECTOR 10
5838	016625	000		.BYTE 0	;TRACK 0
5839	016626	000312		312	;CYLINDER 312
5840					
5841	016630	104417		TLOADRK	;LOAD RK REGS
5842	016632	104424		TWAT32	;WAIT FOR INTERRUPT
5843	016634	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
5844					
5845	016636	104421		TCHKOP	;CHECK OPERATION FOR ANY ERRORS
5846	016640	104004		ERROR 4 ;OR 5 6 7	10 ;REPORT ALL ERRORS
5847	016642	004437	035530	JSR R4,LRLoad	;LOAD "L" REGS
5848	016646	000121		RDDATA	;RDDATA
5849	016650	177400		-400	; -400 WORDS
5850	016652	062414		IBUFF+1000	;IBUFF+1000 IS BUFF ADDRESS
5851	016654	011		.BYTE 11	;SECTOR 11
5852	016655	000		.BYTE 0	;TRACK 0
5853	016656	000312		312	;CYLINDER 312
5854					
5855	016660	104417		TLOADRK	;LOAD RK REGS
5856	016662	104424		TWAT32	;WAIT FOR INTERRUPT
5857	016664	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
5858					
5859	016666	004437	041740	JSR R4,GENCOM	;DATA COMPARE
5860	016672	100000		100000	
5861	016674	001000		1000	;1000 WORDS

5862	016676	000413		BR	25		;NO ERROR-SKIP
5863	016700	104015		ERROR	15		;REPORT FIRST ERROR
5864							
5865	016702	013700	001634	MOV	ERRLMT,R0		;GET ERROR LIMIT
5866	016706	005300		DEC	R0		;DECREMENT COUNT
5867	016710	001406		BEQ	655		;IF ZERO - EXIT
5868	016712	004437	041740	JSR	R4,GENCOM		;CONTINUE DATA COMPARE
5869	016716	040000			40000		
5870	016720	000402		BR	655		;NO MORE ERRORS - EXIT
5871	016722	104016		ERROR	16		;REPORT NEXT ERROR
5872	016724	000770		BR	645		;LOOP
5873	016726						
5874							
5875	016726						
5876							
5877							
5878							
5879							
5880							
5881							
5882							
5883							
5884							
5885	016726	000004					
5886	016730	012737	000012 001262				
5887	016736	104416					
5888	016740	104003					
5889							
5890	016742	004437	035530	JSR	R4,LRLoad		;LOAD "L" REGS
5891	016746	000123		WRDATA			;WRDATA
5892	016750	177000		-1000			; -1000 WORDS
5893	016752	063414		OBUFF			;OBUFF IS BUFF ADDRESS
5894	016754	025		.BYTE	25		;SECTOR 25
5895	016755	000		.BYTE	0		;TRACK 0
5896	016756	000312		312			;CYLINDER 312
5897							
5898	016760	004437	041740	JSR	R4,GENCOM		;GENERATE DATA
5899	016764	000003		3			;PATTERN 3
5900	016766	001000		1000			;1000 WORDS
5901							
5902	016770	104417		TLOADRK			;LOAD RK REGS
5903	016772	104430		TWAIT96			;WAIT FOR INTERRUPT
5904	016774	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
5905							
5906	016776	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
5907	017000	104004		ERROR	4 :OR 5 6 7 10		;REPORT ALL ERRORS
5908							
5909							
5910	017002	004437	041740	JSR	R4,GENCOM		
5911	017006	002007		2007			
5912	017010	001000		1000			
5913							
5914	017012	004437	035530	JSR	R4,LRLoad		;LOAD "L" REGS
5915	017016	000121		RDDATA			;RDDATA
5916	017020	177400		-400			; -400 WORDS
5917	017022	061414		IBUFF			;IBUFF IS BUFF ADDRESS

645:

655:

25:

TEST 50

TEST50:

J09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 112
T50 MID-TRANSFER SEEK ON WRITE (PART 1)

SEQ 0112

```

5918 017024 025 .BYTE 25 ;SECTOR 25
5919 017025 000 .BYTE 0 ;TRACK 0
5920 017026 000312 312 ;CYLINDER 312
5921
5922 017030 104417 TLOADRK ;LOAD RK REGS
5923 017032 104425 TWAT48 ;WAIT FOR INTERRUPT
5924 017034 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5925
5926 017036 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5927 017040 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5928
5929 017042 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGS
5930 017046 000121 RDATA ;RDATA
5931 017050 177400 -400 ;-400 WORDS
5932 017052 062414 IBUFF+1000 ;IBUFF+1000 IS BUFF ADDRESS
5933 017054 000 .BYTE 0 ;SECTOR 0
5934 017055 001 .BYTE 1 ;TRACK 1
5935 017056 000312 312 ;CYLINDER 312
5936
5937 017060 104417 TLOADRK ;LOAD RK REGS
5938 017062 104425 TWAT48 ;WAIT FOR INTERRUPT
5939 017064 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
5940
5941 017066 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
5942 017070 104004 ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
5943
5944 017072 004437 041740 JSR R4,GENCOM ;COMPARE DATA
5945 017076 100000 100000 ;1000 WORDS
5946 017100 001000 1000 ;NO ERRORS-SKIP
5947 017102 000413 BR 15 ;REPORT FIRST ERROR
5948 017104 104015 ERROR 15
5949
5950 017106 013700 001634 MOV ERLMT,R0 ;GET ERROR LIMIT
5951 017112 005300 645: DEC R0 ;DECREMENT COUNT
5952 017114 001406 BEQ 655 ;IF ZERO - EXIT
5953 017116 004437 041740 JSR R4,GENCOM ;CONTINUE DATA COMPARE
5954 017122 040000 40000
5955 017124 000402 BR 655 ;NO MORE ERRORS - EXIT
5956 017126 104016 ERROR 16 ;REPORT NEXT ERROR
5957 017130 000770 BR 645 ;LOOP
5958 017132
5959
5960 017132 15:
5961 *****
5962 *TEST 51 MID-TRANSFER SEEK ON WRITE (PART 2)
5963 *
5964 * ISSUE A WRITE DATA OF 1000 WORDS TO CYLINDER 312,
5965 * TRACK 2, SECTOR 25. READ DATA BACK ONE SECTOR
5966 * AT A TIME AND MAKE SURE A MID-TRANSFER SEEK
5967 * DID TAKE PLACE.
5968 *
5969 *****
5970 017132 000004 ST51: SCOPE
5971 017134 012737 000012 001262 MOV #10.,STIMES ;DO 10. ITERATIONS
5972 017142 104416 TSSINIT ;CLEAR SUBSYSTEM
5973 017144 104003 ERROR 3 ;BAD INIT ERROR

```

K09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 113
 DZRGKD.P11 31-JAN-77 16:26 T51 MID-TRANSFER SEEK ON WRITE (PART 2)

SEQ 0113

5974						
5975	017146	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
5976	017152	000123		WRDATA		;WRDATA
5977	017154	177000		-1000		; -1000 WORDS
5978	017156	063414		IBUFF		;IBUFF IS BUFF ADDRESS
5979	017160	025		.BYTE	25	;SECTOR 25
5980	017161	002		.BYTE	2	;TRACK 2
5981	017162	000312		312		;CYLINDER 312
5982						
5983	017164	004437	041740	JSR	R4,GENCOM	;GENERATE DATA
5984	017170	000004		4		;PATTERN 4
5985	017172	001000		1000		;1000 WORDS
5986						
5987	017174	104417		TLOADRK		;LOAD RK REGS
5988	017176	104430		TWAT96		;WAIT FOR INTERRUPT
5989	017200	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
5990				A CYLINDER ERROR REPORTED AT THE END OF THE OPERATION INDICATES A		
5991				MID-TRANSFER SEEK DID NOT OCCUR.		
5992	017202	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
5993	017204	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
5994						
5995	017206	004437	041740	JSR	R4,GENCOM	;CLEAR IBUFF TO ALL ONES
5996	017212	002007		2007		
5997	017214	001000		1000		
5998						
5999	017216	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
6000	017222	000121		RDDATA		;RDDATA
6001	017224	177400		-400		; -400 WORDS
6002	017226	061414		IBUFF		;IBUFF IS BUFF ADDRESS
6003	017230	025		.BYTE	25	;SECTOR 25
6004	017231	002		.BYTE	2	;TRACK 2
6005	017232	000312		312		;CYLINDER 312
6006						
6007	017234	104417		TLOADRK		;LOAD RK REGS
6008	017236	104425		TWAT48		;WAIT FOR INTERRUPT
6009	017240	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
6010						
6011	017242	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
6012	017244	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
6013						
6014	017246	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
6015	017252	000121		RDDATA		;RDDATA
6016	017254	177400		-400		; -400 WORDS
6017	017256	062414		IBUFF+1000		;IBUFF+1000 IS BUFF ADDRESS
6018	017260	000		.BYTE	0	;SECTOR 0
6019	017261	000		.BYTE	0	;TRACK 0
6020	017262	000313		313		;CYLINDER 313
6021						
6022	017264	104417		TLOADRK		;LOAD RK REGS
6023	017266	104425		TWAT48		;WAIT FOR INTERRUPT
6024	017270	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
6025						
6026	017272	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
6027	017274	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
6028						
6029	017276	004437	041740	JSR	R4,GENCOM	;COMPARE DATA

L09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 114
TS1 MID-TRANSFER SEEK ON WRITE (PART 2)

SEQ 0114

6030	017302	100000		100000	
6031	017304	001000		1000	; 1000 WORDS
6032	017306	000413		BR 15	; NO MISCOMPARES-SKIP
6033	017310	104015		ERROR 15	; REPORT 1ST ERROR
6034					
6035	017312	013700	001634	MOV	ERRLMT, R0 ; GET ERROR LIMIT
6036	017316	005300		DEC	R0 ; DECREMENT COUNT
6037	017320	001406		BEQ	65\$; IF ZERO - EXIT
6038	017322	004437	041740	JSR	R4, GENCOM ; CONTINUE DATA COMPARE
6039	017326	040000		40000	
6040	017330	000402		BR 65\$	; NO MORE ERRORS - EXIT
6041	017332	104016		ERROR 16	; REPORT NEXT ERROR
6042	017334	000770		BR 64\$	; LOOP
6043	017336				
6044					
6045	017336			1\$:	
6046				*****	
6047				*TEST 52	TWO SECTOR READ DATA (PART 1)
6048				*	
6049				*	ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,
6050				*	TRACK 0, SECTOR 0, VERIFY THAT CORRECT DATA IS
6051				*	READ.
6052				*	
6053				*	NOTE: TWO SECTOR WRITE DATA (PART 1) MUST BE
6054				*	EXECUTED BEFORE THIS TEST.
6055				*	
6056				*****	
6057	017336	000004		†ST52:	SCOPE
6058	017340	012737	000012 001262	MOV	#10., \$TIMES ; DO 10. ITERATIONS
6059	017346	104416		TSSINIT	; CLEAR SUBSYSTEM
6060	017350	104003		ERROR 3	; BAD INIT ERROR
6061					
6062					; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA (PART 1)
6063					
6064					; GENERATE SAME DATA AS USED IN TWO SECTOR WRITE DATA PART 1
6065	017352	004437	041740	JSR	R4, GENCOM ; GENERATE DATA
6066	017356	000015		15	; PATTERN 15
6067	017360	001000		1000	; 1000 WORDS
6068					
6069	017362	004437	041740	JSR	R4, GENCOM ; CLEAR Ibuff TO ALL ONES
6070	017366	002007		2007	
6071	017370	001000		1000	
6072					
6073	017372	004437	035530	JSR	R4, LRLOAD ; LOAD "L" REGS
6074	017376	000121		RDDATA	; RDDATA
6075	017400	177000		-1000	; -1000 WORDS
6076	017402	061414		IBUFF	; Ibuff IS BUFF ADDRESS
6077	017404	000		.BYTE 0	; SECTOR 0
6078	017405	000		.BYTE 0	; TRACK 0
6079	017406	000312		312	; CYLINDER 312
6080					
6081	017410	104417		TLOADRK	; LOAD RK REGS
6082	017412	104430		TWAIT96	; WAIT FOR INTERRUPT
6083	017414	104002		ERROR 2	; TO SLOW/NOT COMPLETE ERROR
6084					
6085	017416	104421		TCHKOP	; CHECK OPERATION FOR ANY ERRORS

M09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 115
T52 TWO SECTOR READ DATA (PART 1)

SEQ 0115

6086	017420	104004		ERROR	4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6087					
6088	017422	004437	041740	JSR	R4,GENCOM ;COMPARE DATA
6089	017426	100000		100000	
6090	017430	001000		1000	;1000 WORDS
6091	017432	000413		BR	15 ;NO MISCOMPARES-SKIP
6092	017434	104015		ERROR	15
6093					
6094	017436	013700	001634	MOV	ERRLMT,R0 ;GET ERROR LIMIT
6095	017442	005300		DEC	R0 ;DECREMENT COUNT
6096	017444	001406		BEQ	65\$;IF ZERO - EXIT
6097	017446	004437	041740	JSR	R4,GENCOM ;CONTINUE DATA COMPARE
6098	017452	040000		40000	
6099	017454	000402		BR	65\$;NO MORE ERRORS - EXIT
6100	017456	104016		ERROR	16 ;REPORT NEXT ERROR
6101	017460	000770		BR	64\$;LOOP
6102	017462				
6103					
6104	017462				
6105				1\$:	
6106				*****	
6107				TEST 53	TWO SECTOR READ DATA (PART 2)
6108				*****	
6109				ISSUE A READ DATA OF 1000 WORDS TO CYLINDER 312,	
6110				TRACK 0, SECTOR 23. VERIFY THAT CORRECT DATA IS	
6111				READ AND A MID-TRANSFER SEEK DOES NOT OCCUR.	
6112				*****	
6113				NOTE: TWO SECTOR WRITE DATA (PART 2) MUST BE	
6114				EXECUTED BEFORE THIS TEST.	
6115				*****	
6116	017462	000004		TEST53: SCOPE	
6117	017464	012737	000012 001262	MOV	#10.,\$TIMES ;DO 10. ITERATIONS
6118	017472	104416		TSSINIT	;CLEAR SUBSYSTEM
6119	017474	104003		ERROR	3 ;BAD INIT ERROR
6120					
6121					;GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 2)
6122					
6123	017476	004437	041740	JSR	R4,GENCOM ;GENERATE DATA
6124	017502	000016		16	;PATTERN 16
6125	017504	001000		1000	;1000 WORDS
6126					
6127	017506	004437	041740	JSR	R4,GENCOM ;CLEAR Ibuff TO ALL ONES
6128	017512	002007		2007	
6129	017514	001000		1000	
6130	017516	004437	035530	JSR	R4,LRLOAD ;LOAD "L" REGS
6131	017522	000121		RDDATA	;RDDATA
6132	017524	177000		-1000	; -1000 WORDS
6133	017526	061414		IBUFF	;IBUFF IS BUFF ADDRESS
6134	017530	023		.BYTE	23 ;SECTOR 23
6135	017531	000		.BYTE	0 ;TRACK 0
6136	017532	000312		312	;CYLINDER 312
6137					
6138	017534	104417		TLOADRK	;LOAD RK REGS
6139	017536	104430		TWAIT96	;WAIT FOR INTERRUPT
6140	017540	104002		ERROR	2 ;TO SLOW/NOT COMPLETE ERROR
6141					

N09

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 116
T53 TWO SECTOR READ DATA (PART 2)

SEQ 0116

6142	017542	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
6143	017544	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
6144						
6145	017546	004437	041740	JSR	R4, GENCOM	;COMPARE DATA
6146	017552	100000		100000		
6147	017554	001000		1000		;1000 WORDS
6148	017556	000413		BR	15	;NO MISCOMPARES-SKIP
6149	017560	104015		ERROR	15	;REPORT 1ST ERROR
6150						
6151	017562	013700	001634	MOV	ERRLMT, R0	;GET ERROR LIMIT
6152	017566	005300		DEC	R0	;DECREMENT COUNT
6153	017570	001406		BEQ	655	;IF ZERO - EXIT
6154	017572	004437	041740	JSR	R4, GENCOM	;CONTINUE DATA COMPARE
6155	017576	040000		40000		
6156	017600	000402		BR	655	;NO MORE ERRORS - EXIT
6157	017602	104016		ERROR	16	;REPORT NEXT ERROR
6158	017604	000770		BR	645	;LOOP
6159	017606					
6160						
6161	017606					
6162				15:		
6163				*****		
6164				TEST 54	TWO SECTOR READ DATA (PART 3)	
6165				*		
6166				*	ISSUE A READ DATA OF 401 WORDS TO CYLINDER 312,	
6167				*	TRACK 0, SECTOR 10. VERIFY THAT ALL 401 WORDS	
6168				*	ARE PLACED IN MEMORY.	
6169				*		
6170				*	NOTE: TWO SECTOR WRITE DATA (PART 3) MUST BE	
6171				*	EXECUTED BEFORE THIS TEST.	
6172				*		
6173	017606	000004		*****		
6174	017610	012737	000012 001262	TEST54:	SCOPE	
6175	017616	104416		MOV	#10., \$TIMES	;DO 10. ITERATIONS
6176	017620	104003		TSSINIT		;CLEAR SUBSYSTEM
6177				ERROR	3	;BAD INIT ERROR
6178						
6179				;	GENERATE SAME DATA AS USED IN TWO SECTOR WRITE (PART 3)	
6180	017622	004437	041740	JSR	R4, GENCOM	;GENERATE DATA
6181	017626	000002		2		;PATTERN 2
6182	017630	000401		401		;401 WORDS
6183						
6184	017632	004437	041740	JSR	R4, GENCOM	;CLEAR Ibuff TO ALL ONES
6185	017636	002007		2007		
6186	017640	001000		1000		
6187						
6188	017642	004437	035530	JSR	R4, LRLOAD	;LOAD "L" REGS
6189	017646	000121		RDDATA		;RDDATA
6190	017650	177377		-401		; -401 WORDS
6191	017652	061414		IBUFF		;IBUFF IS BUFF ADDRESS
6192	017654	010		.BYTE	10	;SECTOR 10
6193	017655	000		.BYTE	0	;TRACK 0
6194	017656	000312		312		;CYLINDER 312
6195						
6196	017660	104417		TLOADRK		;LOAD RK REGS
6197	017662	104430		TWAT96		;WAIT FOR INTERRUPT

B10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 117
T54 TWO SECTOR READ DATA (PART 3)

SEQ 0117

6198	017664	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
6199					
6200	017666	104421		TCHKOP	;CHECK OPERATION FOR ANY ERRORS
6201	017670	104004		ERROR 4	;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6202					
6203	017672	004437	041740	JSR R4,GENCOM	;COMPARE DATA
6204	017676	100000		100000	
6205	017700	000401		401	;401 WORDS
6206	017702	000413		BR 15	;NO MISCOMPARES-SKIP
6207	017704	104015		ERROR 15	;PRINT FIRST ERROR
6208					
6209	017706	013700	001634	MOV ERRLMT,R0	;GET ERROR LIMIT
6210	017712	005300		DEC R0	;DECREMENT COUNT
6211	017714	001406		BEQ 655	;IF ZERO - EXIT
6212	017716	004437	041740	JSR R4,GENCOM	;CONTINUE DATA COMPARE
6213	017722	040000		40000	
6214	017724	000402		BR 655	;NO MORE ERRORS - EXIT
6215	017726	104016		ERROR 16	;REPORT NEXT ERROR
6216	017730	000770		BR 645	;LOOP
6217	017732				
6218	017732				
6219					
6220				655:	
6221				15:	
6222				*****	
6223				TEST 55	MID-TRANSFER SEEK ON READ (PART 1)
6224					
6225					
6226					
6227					
6228					
6229					
6230	017732	000004			
6231	017734	012737	000012 001262	ST55: SCOPE	
6232	017742	104416		MOV #10.,STIMES	;DO 10. ITERATIONS
6233	017744	104003		TSSINIT	;CLEAR SUBSYSTEM
6234				ERROR 3	;BAD INIT ERROR
6235					
6236	017746	004437	041740		GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 1)
6237	017752	000003		JSR R4,GENCOM	;GENERATE DATA
6238	017754	001000		3	;PATTERN 3
6239				1000	;1000 WORDS
6240	017756	004437	041740	JSR R4,GENCOM	;CLEAR Ibuff TO ALL ONES
6241	017762	002007		2007	
6242	017764	001000		1000	
6243					
6244	017766	004437	035530	JSR R4,LRLOAD	;LOAD "L" REGS
6245	017772	000121		RDDATA	;RDDATA
6246	017774	177000		-1000	; -1000 WORDS
6247	017776	061414		IBUFF	;IBUFF IS BUFF ADDRESS
6248	020000	025		.BYTE 25	;SECTOR 25
6249	020001	000		.BYTE 0	;TRACK 0
6250	020002	000312		312	;CYLINDER 312
6251					
6252	020004	104417		TLOADRK	;LOAD RK REGS
6253	020006	104430		TWAT96	;WAIT FOR INTERRUPT

C10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 118
T55 MID-TRANSFER SEEK ON READ (PART 1)

SEQ 0118

6254	020010	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
6255							
6256	020012	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
6257	020014	104004		ERROR	4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
6258							
6259	020016	004437	041740	JSR	R4,GENCOM		;COMPARE DATA
6260	020022	100000		100000			
6261	020024	001000		1000			;1000 WORDS
6262	020026	000413		BR	15		;NO MISCOMPARES-SKIP
6263	020030	104015		ERROR	15		;PRINT FIRST ERROR
6264							
6265	020032	013700	001634	MOV	ERRLMT,R0		;GET ERROR LIMIT
6266	020036	005300		DEC	R0		;DECREMENT COUNT
6267	020040	001406		BEQ	655		;IF ZERO - EXIT
6268	020042	004437	041740	JSR	R4,GENCOM		;CONTINUE DATA COMPARE
6269	020046	040000		40000			
6270	020050	000402		BR	655		;NO MORE ERRORS - EXIT
6271	020052	104016		ERROR	16		;REPORT NEXT ERROR
6272	020054	000770		BR	645		;LOOP
6273	020056						
6274	020056						
6275							
6276							
6277							
6278							
6279							
6280							
6281							
6282							
6283							
6284							
6285							
6286	020056	000004		ST56:	SCOPE		
6287	020060	012737	000012 001262	MOV	#10.,\$TIMES		;DO 10. ITERATIONS
6288	020066	104416		TSSINIT			;CLEAR SUBSYSTEM
6289	020070	104003		ERROR	3		;BAD INIT ERROR
6290							
6291							
6292	020072	004437	041740	;	GENERATE SAME DATA AS USED IN MID TRANSFER SEEK ON WRITE (PART 2)		
6293	020076	000004		JSR	R4,GENCOM		;GENERATE DATA
6294	020100	001000		4			;PATTERN 4
6295				1000			;1000 WORDS
6296	020102	004437	041740	JSR	R4,GENCOM		;CLEAR Ibuff TO ALL ONES
6297	020106	002007		2007			
6298	020110	001000		1000			
6299							
6300	020112	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
6301	020116	000121		RDDATA			;RDDATA
6302	020120	177000		-1000			;1000 WORDS
6303	020122	061414		IBUFF			;IBUFF IS BUFF ADDRESS
6304	020124	025		.BYTE	25		;SECTOR 25
6305	020125	002		.BYTE	2		;TRACK 2
6306	020126	000312		312			;CYLINDER 312
6307							
6308	020130	104417		TLOADRK			;LOAD RK REGS
6309	020132	104430		TWAT96			;WAIT FOR INTERRUPT

D10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 119
T56 MID-TRANSFER SEEK ON READ (PART 2)

SEQ 0119

6310	020134	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
6311					
6312	020136	104421		TCHKOP	;CHECK OPERATION FOR ANY ERRORS
6313	020140	104004		ERROR 4	;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6314					
6315	020142	004437	041740	JSR R4,GENCOM	;COMPARE DATA
6316	020146	100000		100000	
6317	020150	001000		1000	;1000 WORDS
6318	020152	000413		BR 15	;NO MISCOMPARES-SKIP
6319	020154	104015		ERROR 15	;REPORT FIRST ERROR
6320					
6321	020156	013700	001634	MOV ERLMT,R0	;GET ERROR LIMIT
6322	020162	005300		DEC R0	;DECREMENT COUNT
6323	020164	001406		BEQ 655	;IF ZERO - EXIT
6324	020166	004437	041740	JSR R4,GENCOM	;CONTINUE DATA COMPARE
6325	020172	040000		40000	
6326	020174	000402		BR 655	;NO MORE ERRORS - EXIT
6327	020176	104016		ERROR 16	;REPORT NEXT ERROR
6328	020200	000770		BR 645	;LOOP
6329	020202				
6330	020202				
6331				655:	
6332				15:	
6333				*****	
6334				*TEST 57 CYLINDER ADDRESS OVERFLOW (PART 1)	
6335				*	
6336				* ISSUE A READ DATA OF 400 WORDS TO CYLINDER 632,	
6337				* TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS	
6338				* OVERFLOW ERROR DOES NOT OCCUR.	
6339	020202	000004		*****	
6340	020204	012737	000012 001262	†ST57: SCOPE	
6341	020212	104416		MOV #10.,\$TIMES	;DO 10. ITERATIONS
6342	020214	104003		TSSINIT	;CLEAR SUBSYSTEM
6343				ERROR 3	;BAD INIT ERROR
6344	020216	004437	035530	JSR R4,LRLOAD	;LOAD "L" REGS
6345	020222	000121		R0DATA	;R0DATA
6346	020224	177400		-400	;400 WORDS
6347	020226	061414		IBUFF	;IBUFF IS BUFF ADDRESS
6348	020230	025		.BYTE 25	;SECTOR 25
6349	020231	002		.BYTE 2	;TRACK 2
6350	020232	000632		632	;CYLINDER 632
6351					
6352	020234	104417		TLOADRK	;LOAD RK REGS
6353	020236	104434		TWAIT159	;WAIT FOR INTERRUPT
6354	020240	104002		ERROR 2	;TO SLOW/NOT COMPLETE ERROR
6355					
6356	020242	104421		TCHKOP	;CHECK OPERATION FOR ANY ERRORS
6357	020244	104004		ERROR 4	;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6358				*****	
6359				*TEST 60 CYLINDER ADDRESS OVERFLOW (PART 2)	
6360				*	
6361				* ISSUE A READ DATA OF 401 WORDS TO CYLINDER 632,	
6362				* TRACK 2, SECTOR 25. MAKE SURE CYLINDER ADDRESS	
6363				* OVERFLOW ERROR DOES OCCUR.	
6364				*	
6365				*****	

E10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 120
T60 CYLINDER ADDRESS OVERFLOW (PART 2)

SEQ 0120

6366	020246	000004			TST60: SCOPE	
6367	020250	012737	000012	001262	MOV #10., \$TIMES	::DO 10. ITERATIONS
6368	020256	104416			TSSINIT	::CLEAR SUBSYSTEM
6369	020260	104003			ERROR 3	::BAD INIT ERROR
6370						
6371	020262	004437	035530		JSR R4, LRLOAD	::LOAD "L" REGS
6372	020266	000121			RDDATA	::RDDATA
6373	020270	177377			-401	::-401 WORDS
6374	020272	061414			IBUFF	::IBUFF IS BUFF ADDRESS
6375	020274	025			.BYTE 25	::SECTOR 25
6376	020275	002			.BYTE 2	::TRACK 2
6377	020276	000632			632	::CYLINDER 632
6378						
6379	020300	104417			TLOADRK	::LOAD RK REGS
6380	020302	104434			TWAT159	::WAIT FOR INTERRUPT
6381	020304	104002			ERROR 2	::TO SLOW/NOT COMPLETE ERROR
6382						
6383	020306	104422			TCHKWE	::CHECK OPERATION WITH EXPECTED ERROR
6384	020310	001000			COERR	::CYLINDER OVERFLOW
6385	020312	000000			0	
6386	020314	000000			0	
6387	020316	104004			ERROR 4; OR 5,6,7	::REPORT ANY DISCREPANCIES

.SBTTL **18 BIT DATA TRANSFER TESTS

```

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

```

6401	020320	000004			TST61: SCOPE	
6402	020322	012737	000012	001262	MOV #10., \$TIMES	::DO 10. ITERATIONS
6403	020330	012737	000312	001676	MOV #312, REFM	::SET REFORMAT SWITCH
6404	020336	012737	020346	001110	MOV #15, \$LPERR	::SET LOCAL LOOP ON ERROR
6405	020344	005001			CLR R1	::CLEAR R1 FOR TRACK COUNTER
6406	020346					
6407	020346	104416			15: TSSINIT	::CLEAR SUBSYSTEM
6408	020350	104003			ERROR 3	::BAD INIT ERROR
6409	020352	012737	010127	001600	MOV #WRHEAD!CFMT, L.CS1	::SET UP FOR WRITE HEADER 24(8) SECTOR MODE
6410	020360	012737	001626	001610	MOV DRVNUM, L.CS2	::SET DRIVE NUMBER
6411	020366	012737	000074	001602	MOV #74, L.WC	::SET WORD COUNT
6412	020374	110137	001607		MOV R1, L.DT	::LOAD TRACK ADDRESS
6413	020400	012737	063414	001604	MOV #0BUFF, L.BA	::SET BUS ADDRESS
6414	020406	012737	000312	001614	MOV #312, L.DCYL	::CYLINDER ADDRESS
6415						
6416	020414	004437	041740		JSR R4, GENCOM	::GENERATE HEADER
6417	020420	001200			1200	::INCLUDE BAD SECTOR BITS
6418						
6419	020422	104417			TLOADRK	::LOAD RK REGS
6420	020424	104434			TWAT159	::WAIT FOR INTERRUPT
6421	020426	104002			ERROR 2	::TO SLOW/NOT COMPLETE ERROR

F10

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZR6KD.P11 31-JAN-77 16:26

 MACY11 27(1006) 31-JAN-77 16:34 PAGE 121
 T61 FORMAT IN 24 SECTOR FORMAT

SEQ 0121

```

6422
6423 020430 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
6424 020432 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
6425
6426 020434 104415      SCOP1          ;LOCAL LOOP ON ERROR TO 1$
6427
6428 020436 005701      TST R1          ;R1 POINTING TO TRACK 0
6429 020440 001002      BNE 2$          ;NO-SKIP
6430 020442 005201      INC R1          ;BUMP TO TRACK 1
6431 020444 000740      BR 1$          ;LOOP
6432 020446 012737 020456 001110 2$: MOV 3$,SLPERR ;SET LOCAL LOOP ON ERROR
6433 020454 005001      CLR R1          ;CLEAR TRACK POINTER
6434
6435 020456 104416      TSSINIT         ;CLEAR SUBSYSTEM
6436 020460 104003      ERROR 3          ;BAD INIT ERROR
6437 020462 012737 010125 001600      MOV 8RDHEAD:CFMT,L.CS1 ;LOAD READ 24(8) SECTOR FORMAT
6438 020470 013737 001626 001610      MOV DRVNUM,L.CS2 ;LOAD DRIVE NUMBER
6439 020476 110137 001607      MOV R1,L.DT ;LOAD TRACK
6440 020502 012737 000312 001614      MOV 312,L.DCYL ;LOAD CYLINDER
6441
6442 020510 004437 036050      JSR R4,RDSTHD ;GO READ STANDARD HEADER
6443 020514 104421      TCHKOP         ;RETURN IF CERR W/O DATA LATE SET
6444 020516 104004      ERROR 4: OR 5,6,7 ;REPORT ALL OTHER ERRORS
6445 020520 104013      ERROR 13        ;RAT DATA LATE
6446 020522 104002      ERROR 2          ;REPORT "OPERATION TOO SLOW" OR "HEADER
6447                                     ;D NOT FOUND" MESSAGE
6448
6449 020524 104415      SCOP1          ;LOCAL LOOP TO 3$ ON ERROR
6450 020526 004437 041740      JSR R4,GENCOM ;GENERATE & COMPARE HEADERS
6451 020532 101200      101200          ;INCLUDING BAD SECTOR LISTS
6452 020534 000413      BR 4$          ;NO MISCOMPARES-SKIP
6453 020536 104015      ERROR 15        ;REPORT FIRST MISCOMPARE
6454
6455 020540 013700 001634      MOV ERRLMT,R0 ;GET ERROR LIMIT
6456 020544 005300      DEC R0          ;DECREMENT COUNT
6457 020546 001406      BEQ 65$        ;IF ZERO - EXIT
6458 020550 004437 041740      JSR R4,GENCOM ;CONTINUE DATA COMPARE
6459 020554 040000      40000          ;
6460 020556 000402      BR 65$        ;NO MORE ERRORS - EXIT
6461 020560 104016      ERROR 16        ;REPORT NEXT ERROR
6462 020562 000770      BR 64$        ;LOOP
6463
6464
6465 020564 104415      4$: SCOP1          ;LOCAL LOOP TO 3$
6466 020566 005701      TST R1          ;POINTING TO TRACK 1
6467 020570 001002      BNE 5$          ;NO-EXIT
6468 020572 005201      INC R1          ;BUMP TO TRACK 1
6469 020574 000730      BR 3$          ;LOOP
6470
6471 020576      5$:
6472      ;*****
6473      ;TEST 62 24 SECTOR FORMAT DATA TRANSFER (PART 1)
6474      ;
6475      ;
6476      ; ISSUE A WRITE DATA OF 400 WORDS IN 24 SECTOR FORMAT
6477      ; TO CYLINDER 312, TRACK 0, SECTOR 0. READ SECTOR BACK
        ; AND MAKE SURE IT IS CORRECT.

```

G10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 122
T62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

SEQ 0122

```

6478
6479
6480 020576 000004
6481 020600 012737 000012 001262
6482 020606 012737 000312 001676
6483 020614 104416
6484 020616 104003
6485
6486 020620 004437 041740
6487 020624 000013
6488 020626 000400
6489
6490 020630 004437 041740
6491 020634 002007
6492 020636 000400
6493
6494 020640 004437 035530
6495 020644 010123
6496 020646 177400
6497 020650 063414
6498 020652 000
6499 020653 000
6500 020654 000312
6501
6502 020656 104417
6503 020660 104430
6504 020662 104002
6505
6506 020664 104421
6507 020666 104004
6508
6509 020670 004437 035530
6510 020674 010121
6511 020676 177400
6512 020700 061414
6513 020702 000
6514 020703 000
6515 020704 000312
6516
6517 020706 104417
6518 020710 104424
6519 020712 104002
6520
6521 020714 104421
6522 020716 104004
6523
6524 020720 004437 041740
6525 020724 100000
6526 020726 000400
6527 020730 000413
6528 020732 104015
6529
6530 020734 013700 001634
6531 020740 005300
6532 020742 001406
6533 020744 004437 041740

;#
;*****
;T62: SCOPE
;MOV #10,STIMES ;DO 10 ITERATIONS
;MOV #312,REFMT ;SET REFORMAT SWITCH
;TSSINIT ;CLEAR SUBSYSTEM
;ERROR 3 ;BAD INIT ERROR
;
;JSR R4,GENCOM ;GENERATE DATA
;13 ;PATTERN 13
;400 ;400 WORDS
;
;JSR R4,GENCOM ;CLEAR Ibuff TO ALL ONES
;2007
;400
;
;JSR R4,LRLoad ;LOAD "L" REGS
;WRDATA!CFMT ;WRDATA!CFMT
;-400 ;-400 WORDS
;OBUFF ;OBUFF IS BUff ADDRESS
;.BYTE 0 ;SECTOR 0
;.BYTE 0 ;TRACK 0
;312 ;CYLINDER 312
;
;TLOADRK ;LOAD RK REGS
;TWAT96 ;WAIT FOR INTERRUPT
;ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
;
;TCHKOP ;CHECK OPERATION FOR ANY ERRORS
;ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
;
;JSR R4,LRLoad ;LOAD "L" REGS
;RDATA!CFMT ;RDATA!CFMT
;-400 ;-400 WORDS
;IBUFF ;IBUFF IS BUff ADDRESS
;.BYTE 0 ;SECTOR 0
;.BYTE 0 ;TRACK 0
;312 ;CYLINDER 312
;
;TLOADRK ;LOAD RK REGS
;TWAT32 ;WAIT FOR INTERRUPT
;ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
;
;TCHKOP ;CHECK OPERATION FOR ANY ERRORS
;ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
;
;JSR R4,GENCOM ;COMPARE DATA
;100000
;400 ;400 WORDS
;BR 15 ;NO MISCOMPARES-SKIP
;ERROR 15 ;REPORT 1ST ERROR
;
;MOV ERRLMT,R0 ;GET ERROR LIMIT
;DEC R0 ;DECREMENT COUNT
;BEQ 655 ;IF ZERO - EXIT
;JSR R4,GENCOM ;CONTINUE DATA COMPARE

```

645:

H10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 123
T62 24 SECTOR FORMAT DATA TRANSFER (PART 1)

SEQ 0123

```

6534 020750 040000          40000
6535 020752 000402          BR      655          ;NO MORE ERRORS - EXIT
6536 020754 104016          ERROR 16          ;REPORT NEXT ERROR
6537 020756 000770          BR      645          ;LOOP
6538 020760
6539
6540 020760
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554 020760 000004          1S:
6555 020762 012737 000012 001262          ;*****
6556 020770 012737 000312 001676          ;TEST 63      24 SECTOR FORMAT DATA TRANSFER (PART 2)
6557 020776 032737 000100 001664          ;*
6558 021004 001556          ;*      LOAD A LOCATION WITH BAD PARITY.  ISSUE A WRITE DATA OF
6559          ;*      400 WORDS IN 24 SECTOR FORMAT TO CYLINDER 312, TRACK 0,
6560 021006 004437 041740          ;*      SECTOR 0 WITH BUFFER BEGINNING 112 WORDS BEFORE WORD
6561 021012 000007          ;*      WITH BAD PARITY.  MAKE SURE UNIBUS PARITY ERROR DOES NOT SET.
6562 021014 000400          ;*      READ SECTOR BACK AND MAKE SURE IT IS CORRECT.
6563          ;*
6564 021016 032737 002000 001664          ;*      NOTE:  THIS TEST IS EXECUTED ONLY IF MEMORY PARITY
6565 021024 001023          ;*      EXISTS FOR SPECIFIED LOCATION.
6566 021026 012746 000340          ;*****
6567 021032 012746 021040          ;TEST 63:  SCOPE
6568 021036 000002          ;MOV      #10, $TIMES          ;DO 10. ITERATIONS
6569 021040          ;MOV      #312, REFM      ;SET REFORMAT SWITCH
6570          ;BIT      #PARPRE, OPTFLG      ;PARITY OPTION PRESENT?
6571 021040 012777 000004 160624          ;BEQ      45          ;YES-SKIP
6572 021046 012737 100200 063640          1S:      JSR      R4, GENCOM      ;GENERATE DATA
6573 021054 012777 000001 160610          ;7          ;PATTERN 7
6574          ;400          ;400 WORDS
6575 021062 013746 001622          ;BIT      #CP1170, OPTFLG      ;TEST IF 11/70
6576 021066 012746 021074          ;BNE      11S          ;YES - SKIP
6577 021072 000002          ;MOV      #PR7, -(SP)      ;SET PRIORITY TO 7
6578 021074          ;MOV      #105, -(SP)      ;SET ADDRESS
6579 021074 104416          ;RTI
6580 021076 104003          ;10S:
6581 021100 004437 035530          ;MOV      #BIT2, CSRPTR      ;SET WRONG PARITY WRITE
6582 021104 010123          ;MOV      #100200, OBUF+224      ;WRITE WITH BAD PARITY
6583 021106 177400          ;MOV      #BIT0, CSRPTR      ;CLEAR WRONG PARITY, SET IE
6584 021110 063414          ;6S:      MOV      RKPRI, -(SP)      ;RESTORE PRIORITY
6585 021112 000          ;MOV      #11S, -(SP)
6586 021113 000          ;RTI
6587 021114 000312          ;11S:
6588          ;TSSINIT          ;CLEAR SUBSYSTEM
6589 021116 013704 001672          ;ERROR      3          ;BAD INIT ERROR
        ;JSR      R4, LRLOAD          ;LOAD "L" REGS
        ;WRDATA!CFMT          ;WRDATA!CFMT
        ;-400          ;-400 WORDS
        ;OBUF          ;OBUF IS BUFF ADDRESS
        ;.BYTE      0          ;SECTOR 0
        ;.BYTE      0          ;TRACK 0
        ;312          ;CYLINDER 312
        ;MOV      CSRPTR, R4          ;LOAD R4 WITH CSR POINTER

```

6590	021122	005000			CLR	RO	; CLEAR RO FOR COUNTING
6591	021124	012701	001662		MOV	#INTSET, R1	; LOAD R1 WITH POINTER INTERRUPT FLAG
6592	021130	012777	061400	160462	MOV	#SPCHLR, @RKVEC	; SET INT VECTOR FOR RK611
6593	021136	012777	061410	160552	MOV	#SPCPAR, @MMVECA	; AND PARITY ERROR
6594	021144	012737	021342	001264	MOV	#45, \$ESCAPE	; SET ESCAPE FOR ERROR
6595	021152	104417			TLOADRK		; LOAD RK REGS
6596	021154	032737	002000	001664	BIT	#CP1170, OPTFLG	; TEST IF 11/70
6597	021162	001433			BEQ	45\$	; NO - SKIP
6598	021164	012737	000016	177746	MOV	#16, 177746	; SET TO DISABLE CACHE
6599	021172	012777	000000	160510	MOV	#0, @KWLADD	; DISABLE CLOCK INTERRUPTS
6600	021200	012777	000000	157736	MOV	#0, @STKS	; TURN OFF KEYBOARD INTERRUPTS
6601	021206	012714	170000		MOV	#170000, (R4)	; SET TO FORCE PARITY ERROR
6602							
6603							
6604	021212	105711			40\$:	TSTB (R1)	; LOOP TO WAIT FOR INTERRUPT. THE
6605	021214	003005				BGT 43\$	; CODE BETWEEN THE STARS IS SETUP SO ALL
6606	021216	005300				DEC RO	; BYTES HAVE PARITY OF 1, HENCE NO PARITY
6607	021220	100774				BMI 40\$	; TRAPS ON AN 11/70. IF THIS CODE IS CHA-
6608	021222	000240				NOP	; NGED, REMEMBER ALL BYTES MUST HAVE AN
6609	021224	003372				BGT 40\$	; EVEN NUMBER OF BITS.
6610	021226	000240				NOP	
6611	021230	005014			43\$:	CLR (R4)	; CLEAR ERROR FORCE
6612							
6613							
6614	021232	005037	177746			CLR 177746	; ENABLE CACHE
6615	021236	012777	000100	160444		MOV #BIT6, @KWLADD	; ENABLE CLOCK INTERRUPTS
6616	021244	012777	000100	157672		MOV #100, @STKS	; TURN ON KEYBOARD INTERRUPTS
6617	021252	104430			45\$:	TWAT96	; WAIT FOR INTERRUPT
6618	021254	000414				BR 46\$	; TO SLOW/NOT COMPLETE ERROR
6619	021256	032737	002000	001664		BIT #CP1170, OPTFLG	; TEST IF 11/70
6620	021264	001024				BNE 48\$	; YES - SKIP
6621	021266	005077	160400			CLR @CSRPTR	; CLEAR PARITY DETECT
6622	021272	005037	063640			CLR @BUFF+224	; CLEAR BAD PARITY WORD
6623	021276	012777	000001	160366		MOV #1, @CSRPTR	; ENABLE PARITY DETECT
6624	021304	000414				BR 48\$	; SKIP
6625							
6626	021306	032737	002000	001664	46\$:	BIT #CP1170, OPTFLG	; TEST IF 11/70
6627	021314	001007				BNE 47\$	; YES - SKIP
6628	021316	005077	160350			CLR @CSRPTR	; CLEAR PARITY DETECT
6629	021322	005037	063640			CLR @BUFF+224	; CLEAR BAD PARITY WORD
6630	021326	012777	000001	160336		MOV #1, @CSRPTR	; ENABLE PARITY DETECT
6631	021334	104002			47\$:	ERROR 2	; REPORT TO SLOW ERROR
6632							
6633	021336				48\$:		
6634	021336	104421				TCHKOP	; CHECK OPERATION FOR ANY ERRORS
6635	021340	104004				ERROR 4 ; OR 5, 6, 7, 10	; REPORT ALL ERRORS
6636							
6637							
6638							
6639							
6640							
6641	021342	012777	034354	160250	4\$:	MOV #INTHLR, @RKVEC	; RESET INT VECTOR FOR RK611
6642	021350	012777	034364	160340		MOV #PERHLR, @MMVECA	; AND PARITY ERROR
6643	021356	012777	000100	160324		MOV #BIT6, @KWLADD	; ENABLE CLOCK
6644	021364	012777	000100	157552		MOV #100, @STKS	; ENABLE KEYBOARD INTERRUPTS
6645							

J10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 125
T63 24 SECTOR FORMAT DATA TRANSFER (PART 2)

SEQ 0125

```

6646
6647
6648
6649
6650
6651
6652
6653
6654
6655
6656 021372 000004
6657 021374 012737 000012 001262
6658 021402 012737 000312 001676
6659 021410 004737 034434
6660 021414 104416
6661 021416 104003
6662
6663 021420 004437 041740
6664 021424 000015
6665 021426 001000
6666
6667 021430 004437 041740
6668 021434 002007
6669 021436 001000
6670
6671 021440 004437 035530
6672 021444 010123
6673 021446 177000
6674 021450 063414
6675 021452 023
6676 021453 000
6677 021454 000312
6678
6679 021456 104417
6680 021460 104430
6681 021462 104002
6682
6683 021464 104421
6684 021466 104004
6685
6686 021470 004437 035530
6687 021474 010121
6688 021476 177000
6689 021500 061414
6690 021502 023
6691 021503 000
6692 021504 000312
6693
6694 021506 104417
6695 021510 104426
6696 021512 104002
6697
6698 021514 104421
6699 021516 104004
6700
6701 021520 004437 041740

```

```

*****
*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.
*****
T64: SCOPE
MOV      #10.,$TIMES      ;DO 10. ITERATIONS
MOV      #312,REFMT      ;SET REFORMAT SWITCH
JSR      PC,OPTTST        ;SET UP OPTIONS
TSSINIT                      ;CLEAR SUBSYSTEM
ERROR    3                ;BAD INIT ERROR

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

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

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

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

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

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

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

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

JSR      R4,GENCOM        ;COMPARE DATA

```

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRSKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 126
T64 24 SECTOR FORMAT DATA TRANSFER (PART 3)

SEQ 0126

Address	Hex	Dec	Hex	Dec	Hex	Dec	Label	Operation	Comments
6702	021524	100000							100000
6703	021526	001000							1000 WORDS
6704	021530	000413						BR 15	NO MISCOMPARES-SKIP
6705	021532	104015						ERROR 15	REPORT FIRST ERROR
6706									
6707	021534	013700	001634				645:	MOV	ERRLMT, R0
6708	021540	005300						DEC	R0
6709	021542	001406						BEQ	655
6710	021544	004437	041740					JSR	R4, GENCOM
6711	021550	040000							40000
6712	021552	000402						BR	655
6713	021554	104016						ERROR	16
6714	021556	000770						BR	645
6715	021560						655:		NO MORE ERRORS - EXIT
6716									REPORT NEXT ERROR
6717	021560						15:		LOOP
6718									
6719									
6720									
6721									
6722									
6723									
6724									
6725									
6726									
6727									
6728									
6729									
6730									
6731									
6732									
6733									
6734									
6735	021560	000004							
6736	021562	012737	000012	001262					
6737	021570	012737	000312	001676					
6738	021576	104416							
6739	021600	104003							
6740									
6741	021602	004437	035530						
6742	021606	000127							
6743	021610	177676							
6744	021612	063414							
6745	021614	000							
6746	021615	000							
6747	021616	000312							
6748									
6749	021620	004437	041740						
6750	021624	000600							
6751									
6752	021626	042737	040000	063424					
6753	021634	042737	040000	063426					
6754									
6755	021642	104417							
6756	021644	104431							

Address	Offset	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	
---------	--------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

M10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 128
T65 MULTI-SECTOR DATA TRANSFER AND BSE

SEQ 0128

6814	022076	061414		IBUFF		;IBUFF IS BUFF ADDRESS
6815	022100	000		.BYTE	0	;SECTOR 0
6816	022101	000		.BYTE	0	;TRACK 0
6817	022102	000312		312		;CYLINDER 312
6818						
6819	022104	104417		TLOADRK		;LOAD RK REGS
6820	022106	104424		TWAT32		;WAIT FOR INTERRUPT
6821	022110	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
6822						
6823	022112	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
6824	022114	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
6825						
6826	022116	004437	041740	JSR	R4,GENCOM	;COMPARE DATA
6827	022122	100000		100000		
6828	022124	000400		400		;400 WORDS
6829	022126	000413		BR	4\$	;NO MISCOMPARES-EXIT
6830	022130	104015		ERROR	15	;REPORT FIRST ERROR
6831						
6832	022132	013700	001634	MOV	ERRLMT,R0	;GET ERROR LIMIT
6833	022136	005300		DEC	R0	;DECREMENT COUNT
6834	022140	001406		BEO	65\$	;IF ZERO - EXIT
6835	022142	004437	041740	JSR	R4,GENCOM	;CONTINUE DATA COMPARE
6836	022146	040000		40000		
6837	022150	000402		BR	65\$	;NO MORE ERRORS - EXIT
6838	022152	104016		ERROR	16	;REPORT NEXT ERROR
6839	022154	000770		BR	64\$	;LOOP
6840	022156					
6841						
6842	022156	004437	041740	JSR	R4,GENCOM	;CLEAR IBUFF
6843	022162	002007		2007		
6844	022164	001000		1000		
6845						
6846	022166	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
6847	022172	000121		RDDATA		;RDDATA
6848	022174	177000		-1000		; -1000 WORDS
6849	022176	061414		IBUFF		;IBUFF IS BUFF ADDRESS
6850	022200	000		.BYTE	0	;SECTOR 0
6851	022201	000		.BYTE	0	;TRACK 0
6852	022202	000312		312		;CYLINDER 312
6853						
6854	022204	104417		TLOADRK		;LOAD RK REGS
6855	022206	104424		TWAT32		;WAIT FOR INTERRUPT
6856	022210	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
6857						
6858	022212	104422		TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
6859	022214	000000		0		
6860	022216	000100		BSERR		;BAD SECTOR ERROR
6861	022220	000000		0		
6862	022222	104004		ERROR	4; OR 5,6,7	;REPORT ALL DISCREPANCIES
6863						
6864	022224	004437	041740	JSR	R4,GENCOM	;COMPARE DATA AGAIN
6865	022230	100000		100000		
6866	022232	000400		400		;400 WORDS
6867	022234	000413		BR	5\$	;NO MISCOMPARES
6868	022236	104015		ERROR	15	;REPORT FIRST ERROR
6869						

N10

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 129
T65 MULTI-SECTOR DATA TRANSFER AND BSE

SEQ 0129

6870 022240 013700 001634
6871 022244 005300
6872 022246 001406
6873 022250 004437 041740
6874 022254 040000
6875 022256 000402
6876 022260 104016
6877 022262 000770
6878 022264
6879
6880 022264
6881
6882
6883
6884
6885
6886
6887
6888
6889 022264 000004
6890 022266 012737 000001 001262
6891 022274 005000
6892 022276 012737 022304 001110
6893
6894 022304
6895 022304 104416
6896 022306 104003
6897
6898 022310 013737 001626 001610
6899 022316 012737 000127 001600
6900 022324 110037 001607
6901 022330 012737 063414 001604
6902 022336 012737 177676 001602
6903 022344 012737 000312 001614
6904
6905 022352 004437 041740
6906 022356 001200
6907
6908 022360 104417
6909 022362 104431
6910 022364 104002
6911
6912 022366 104421
6913 022370 104004
6914
6915 022372 104415
6916
6917 022374 005700
6918 022376 001002
6919 022400 005200
6920 022402 000740
6921 022404 005000
6922 022406 012737 022414 001110
6923
6924 022414
6925 022414 104416

66\$: MOV ERRMT,RO ;GET ERROR LIMIT
DEC RO ;DECREMENT COUNT
BEQ 67\$;IF ZERO - EXIT
JSR R4,GENCOM ;CONTINUE DATA COMPARE
40000
BR 67\$;NO MORE ERRORS - EXIT
ERROR 16 ;REPORT NEXT ERROR
BR 66\$;LOOP

67\$:

5\$:

;TEST 66 FORMAT TST

; 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.

;ST66: SCOPE
MOV #1,STIME\$;DO 1 ITERATION
CLR RO ;CLEAR TRACK COUNTER
MOV #15,\$LPERR ;SET LOCAL LOOP

1\$: TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

MOV DRVNUM,L.CS2 ;LOAD DRIVE NUMBER
MOV #WHEAD,L.CS1 ;LOAD WRITE HEADER
MOV# RO,L.DT ;LOAD DESIRED TRACK FROM TRACK COUNTER
MOV #08UFF,L.BA ;LOAD BUS ADDRESS
MOV #-102,L.WC ;WORD COUNT
MOV #312,L.DCYL ;CYLINDER

JSR R4,GENCOM ;BUILD HEADER
1200 ;WITH BSE FLAGGED

TLOADRK ;LOAD RK REGS
THAT112 ;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

SCOPI ;LOCAL LOOP TO 1\$

TST RO ;RO AT ZERO?
BNE 2\$;NO-EXIT
INC RO ;BUMP COUNTER
BR 1\$;LOOP

2\$: CLR RO ;CLEAR SECTOR COUNTER
MOV #113,\$LPERR ;SET LOCAL LOOP ON ERROR

113\$: TSSINIT ;CLEAR SUBSYSTEM

B11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 130
T66 FORMAT TEST

SEQ 0130

```

6926 022416 104003
6927 022420
6928 022420 004437 035530
6929 022424 000121
6930 022426 177400
6931 022430 061414
6932 022432 000
6933 022433 000
6934 022434 000312
6935
6936 022436 110037 001606
6937
6938 022442 104417
6939 022444 104424
6940 022446 104002
6941
6942 022450 104421
6943 022452 104004
6944
6945 022454 104415
6946
6947 022456 022700 000024
6948 022462 001402
6949 022464 005200
6950 022466 000754
6951
6952 022470 005037 001676
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964 022474 000004
6965 022476 012737 000012 001262
6966 022504 104416
6967 022506 104003
6968
6969 022510 004437 035530
6970 022514 000123
6971 022516 177400
6972 022520 063414
6973 022522 000
6974 022523 000
6975 022524 000312
6976
6977 022526 004437 041740
6978 022532 000002
6979 022534 000400
6980
6981 022536 104417

3$: ERROR 3 ;BAD INIT ERROR
JSR R4,LRLOAD ;LOAD "L" REGS
R0DATA ;R0DATA
-400 ; -400 WORDS
IBUFF ;IBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

MOV8 R0,L.DS ;LOAD SECTOR COUNTER INTO DESIRED SECTOR

TLOADRK ;LOAD RK REGS
THAT32 ;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

SCOP1 ;LOCAL LOOP TO 3$ ON ERROR

CMP #24,R0 ;LAST SECTOR READ?
BEQ 4$ ;YES-EXIT
INC R0 ;BUMP SECTOR COUNTER
BR 3$ ;LOOP

4$: CLR REFORMAT ;CLEAR REFORMAT SWITCH

.SBTTL **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.
;*****
TST67: SCOPE
MOV #10.,$TIMES ;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

JSR R4,LRLOAD ;LOAD "L" REGS
R0DATA ;R0DATA
-400 ; -400 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
312 ;CYLINDER 312

JSR R4,GENCOM ;GENERATE DATA
2 ;PATTERN 2
400 ;400 WORDS

TLOADRK ;LOAD RK REGS

```

C11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 131
T67 WRITE-CHECK WITH NO ERROR

SEQ 0131

```

6982 022540 104430      TWAT96      ;WAIT FOR INTERRUPT
6983 022542 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
6984
6985 022544 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
6986 022546 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
6987
6988 022550 004437 035530 JSR R4,LRLOAD ;LOAD "L" REGS
6989 022554 000131      WRCHK      ;WRCHK
6990 022556 177400      -400      ;-400 WORDS
6991 022560 063414      OBUF      ;OBUF IS BUFF ADDRESS
6992 022562 000      .BYT 0      ;SECTOR 0
6993 022563 000      .BYTE 0      ;TRACK 0
6994 022564 000312      312      ;CYLINDER 312
6995
6996 022566 104417      TLOADRK     ;LOAD RK REGS
6997 022570 104424      TWAT32     ;WAIT FOR INTERRUPT
6998 022572 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
6999
7000 022574 104421      TCHKOP     ;CHECK OPERATION FOR ANY ERRORS
7001 022576 104004      ERROR 4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7002
7003 *****
7004 *TEST 70 WRITE CHECK ERROR (PART 1)
7005 *
7006 * WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH ALL ZEROES.
7007 * WRITE CHECK CYLINDER 312, TRACK 0, SECTOR 0 WITH SAME
7008 * DATA EXCEPT WORD 110 HAS ONE OF THE FOLLOWING
7009 * CONFIGURATIONS:
7010 *
7011 * 000001 000020 000400 010000
7012 * 000002 000040 001000 020000
7013 * 000004 000100 002000 040000
7014 * 000010 000200 004000 100000
7015 *
7016 * MAKE SURE WRITE CHECK ERROR SET FOR EACH
7017 * OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
7018 * AND WORD COUNT IS CORRECT.
7019 *
7020 *****
7021 *ST70: SCOPE
7022 *MOV #10,STIMES ;DO 10. ITERATIONS
7023 *MOV #BIT0,R0 ;SET LO ORDER BIT IN R0 FOR
7024 ;CAUSING WRITE CHECK ERROR
7025 TSSINIT ;CLEAR SUBSYSTEM
7026 ERROR 3 ;BAD INIT ERROR
7027 JSR R4,GENCOM ;GENERATE DATA, ALL 0'S
7028 1
7029 400
7030 JSR R4,LRLOAD ;LOAD "L" REGS
7031 WRDATA ;WRDATA
7032 -400 ;-400 WORDS
7033 OBUF ;OBUF IS BUFF ADDRESS
7034 .BYT 0 ;SECTOR 0
7035 .BYTE 0 ;TRACK 0
7036 312 ;CYLINDER 312
7037

```

D11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 132
T70 WRITE CHECK ERROR (PART 1)

SEQ 0132

7038	022646	104417			TLOADRK		;LOAD RK REGS
7039	022650	104430			TWAT96		;WAIT FOR INTERRUPT
7040	022652	104002			ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7041							
7042	022654	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7043	022656	104004			ERROR 4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7044							
7045	022660	004437	035530		JSR R4,LRLOAD		;LOAD "L" REGS
7046	022664	000131			WRTCHK		;WRTCHK
7047	022666	177400			-400		; -400 WORDS
7048	022670	063414			OBUFF		;OBUFF IS BUFF ADDRESS
7049	022672	000			.BYTE 0		;SECTOR 0
7050	022673	000			.BYTE 0		;TRACK 0
7051	022674	000312			312		;CYLINDER 312
7052							
7053	022676	104417			TLOADRK		;LOAD RK REGS
7054	022700	104424			TWAT32		;WAIT FOR INTERRUPT
7055	022702	104002			ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7056							
7057	022704	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7058	022706	104004			ERROR 4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7059							
7060	022710	104415			SCOPI		;LOCAL LOOP ON WRITE CHECK
7061	022712	012737	022720	001110	MOV #15,SLPERR		;SET LOCAL LOOP
7062	022720	010037	063634	15:	MOV R0,OBUFF+220		;CAUSE ERROR BIT IN BUFFER
7063	022724	104416			TSSINIT		;CLEAR SUBSYSTEM
7064	022726	104003			ERROR 3		;BAD INIT ERROR
7065	022730	004437	035530		JSR R4,LRLOAD		;LOAD "L" REGS
7066	022734	000131			WRTCHK		;WRTCHK
7067	022736	177400			-400		; -400 WORDS
7068	022740	063414			OBUFF		;OBUFF IS BUFF ADDRESS
7069	022742	000			.BYTE 0		;SECTOR 0
7070	022743	000			.BYTE 0		;TRACK 0
7071	022744	000312			312		;CYLINDER 312
7072							
7073	022746	104417			TLOADRK		;LOAD RK REGS
7074	022750	104424			TWAT32		;WAIT FOR INTERRUPT
7075	022752	104002			ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7076							
7077	022754	104422			TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
7078	022756	000000			0		
7079	022760	000004			WCKERR		;WRITE CHECK ERROR
7080	022762	000000			0		
7081	022764	104004			ERROR 4	;OR 5,6,7	;REPORT ALL DISCREPANCIES
7082							
7083	022766	104415			SCOPI		;LOCAL LOOP ON ERROR TO 15
7084							
7085							
7086							
7087							
7088							
7089							
7090							
7091							
7092							
7093							

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

E11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZREKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 133
T70 WRITE CHECK ERROR (PART 1)

SEQ 0133

```

7094
7095 022770 023727 001544 063636      CMP      T.BA,#0BUFF+222 ;CHECK BA HALT AT PROPER PLACE
7096 022776 001416                      BEQ      2$              ;YES-SKIP
7097 023000 101040                      BHI      6$              ;IF TO HI - SKIP
7098 023002 012737 054021 001450      MOV      #EM11,EM10N ;"INCORRECT BA"
7099 023010 012737 063636 001202      MOV      #0BUFF+222,$REG10 ;GOOD VALUE
7100 023016 013737 001544 001204      MOV      T.BA,$REG11 ;BAD VALUE
7101 023024 012737 050203 060770      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7102 023032 104010                      ERROR    10
7103
7104 023034 023727 001542 177511 2$:    CMP      T.WC,#-267      ;CHECK WORD COUNT AT CORRECT VALUE
7105 023042 001461                      BEQ      3$              ;YES-SKIP
7106 023044 101037                      BHI      7$              ;IF HIGHER SKIP
7107 023046 012737 053774 001450      MOV      #EM10,EM10N ;"INCORRECT WC"
7108 023054 012737 050203 060770      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7109 023062 012737 177511 001202      MOV      #-267,$REG10 ;GOOD VALUE
7110 023070 013737 001542 001204      MOV      T.WC,$REG11 ;ERROR VALUE
7111 023076 104010                      ERROR    10
7112 023100 000742                      BR       3$              ;EXIT
7113
7114 023102 023727 001544 063642 6$:    CMP      T.BA,#0BUFF+226 ;TEST IF BA AT HI SIDE
7115 023110 101415                      BLOS    7$              ;YES - SKIP
7116 023112 012737 054021 001450      MOV      #EM11,EM10N ;SET MESSAGE
7117 023120 012737 063642 001202      MOV      #0BUFF+226,$REG10 ;GOOD VALUE
7118 023126 013737 001544 001204      MOV      T.BA,$REG11 ;ERROR VALUE
7119 023134 012737 050203 060770      MOV      #OPER41,DF010A ;"WRITE CHECK ABORTED WITH WCE"
7120 023142 104010                      ERROR    10
7121
7122 023144 023727 001542 177513 7$:    CMP      T.WC,#-265      ;TEST IF WORD COUNT AT HI SIDE
7123 023152 101415                      BLOS    3$              ;YES - SKIP
7124 023154 012737 053774 001450      MOV      #EM10,EM10N ;SET MESSAGE
7125 023162 012737 050203 060770      MOV      #OPER41,DF010A ;"WC ABORTED WITH WCE"
7126 023170 012737 177513 001202      MOV      #-265,$REG10 ;GOOD VALUE
7127 023176 013737 001542 001204      MOV      T.WC,$REG11 ;ERROR VALUE
7128 023204 104010                      ERROR    10
7129
7130 023206 104415                      3$:      SCOP1              ;LOCAL LOOP ON ERROR TO 1$
7131
7132 023210 032700 100000                      BIT      #BIT15,R0      ;BIT 15 SET?
7133 023214 001002                      BNE     4$              ;YES-EXIT
7134 023216 006300                      ASL     R0              ;SHIFT ERROR BIT
7135 023220 000637                      BR      1$              ;LOOP
7136
7137 023222                      4$:
7138 *****
7139 *TEST 71 WRITE CHECK ERROR (PART 2)
7140 *
7141 * WRITE CYLINDER 312, TRACK 0, SECTOR 0 WITH 17777
7142 * IN ALL WORDS. WRITE CHECK CYLINDER 312, TRACK 0,
7143 * SECTOR 0 WITH THE SAME DATA EXCEPT WORD 120 HAS
7144 * ONE OF THE FOLLOWING CONFIGURATIONS:
7145 *
7146 * 177776 177757 177377 167777
7147 * 177775 177737 176777 157777
7148 * 177773 177677 175777 137777
7149 * 177767 177577 173777 077777

```

F11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 134
T71 WRITE CHECK ERROR (PART 2)

SEQ 0134

```

7150
7151
7152
7153
7154
7155
7156 023222 000004
7157 023224 012737 000012 001262
7158 023232 012700 177776
7159
7160 023236 104416
7161 023240 104003
7162 023242 004437 041740
7163 023246 000007
7164 023250 000400
7165
7166 023252 004437 035530
7167 023256 000123
7168 023260 177400
7169 023262 063414
7170 023264 000
7171 023265 000
7172 023266 000312
7173
7174 023270 104417
7175 023272 104430
7176 023274 104002
7177
7178 023276 104421
7179 023300 104004
7180
7181 023302 004437 035530
7182 023306 000131
7183 023310 177400
7184 023312 063414
7185 023314 000
7186 023315 000
7187 023316 000312
7188
7189 023320 104417
7190 023322 104424
7191 023324 104002
7192
7193 023326 104421
7194 023330 104004
7195
7196 023332 104415
7197 023334 012737 023342 001110
7198
7199 023342 010037 063634
7200 023346 104416
7201 023350 104003
7202
7203 023352 004437 035530
7204 023356 000131
7205 023360 177400

```

;*
 ;* MAKE SURE WRITE CHECK ERROR SET FOR EACH
 ;* OF THE CONFIGURATIONS AND THAT THE BUS ADDRESS
 ;* AND WORD COUNT IS CORRECT.
 ;*
 ;*****
 T71: SCOPE
 MOV #10,STIMES ;DO 10. ITERATIONS
 MOV #177776,R0 ;LOAD R0 FOR CAUSING WRITE CHECK ERROR

 TSSINIT ;CLEAR SUBSYSTEM
 ERROR 3 ;BAD INIT ERROR
 JSR R4,GENCOM ;GENERATE DATA
 7 ;ALL 1'S
 400 ;400 WORDS

 JSR R4,LRLoad ;LOAD "L" REGS
 WRDATA ;WRDATA
 -400 ;-400 WORDS
 OBUF ;OBUF IS BUFF ADDRESS
 .BYTE 0 ;SECTOR 0
 .BYTE 0 ;TRACK 0
 312 ;CYLINDER 312

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

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

 JSR R4,LRLoad ;LOAD "L" REGS
 WRTCHK ;WRTCHK
 -400 ;-400 WORDS
 OBUF ;OBUF IS BUFF ADDRESS
 .BYTE 0 ;SECTOR 0
 .BYTE 0 ;TRACK 0
 312 ;CYLINDER 312

 TLOADRK ;LOAD RK REGS
 TWAT32 ;WAIT FOR INTERRUPT
 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR

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

 SCOP1 ;LOCAL LOOP TO START OF TEST
 MOV #15,\$LPERR ;SET LOCAL LOOP

 15: MOV R0,OBUF+220 ;PUT WORD I' BUFF TO CAUSE WCE
 TSSINIT ;CLEAR SUBSYSTEM
 ERROR 3 ;BAD INIT ERROR

 JSR R4,LRLoad ;LOAD "L" REGS
 WRTCHK ;WRTCHK
 -400 ;-400 WORDS

G11

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZR6KD.P11 31-JAN-77 16:26

 MACY11 27(1006) 31-JAN-77 16:34 PAGE 135
 T71 WRITE CHECK ERROR (PART 2)

SEQ 0135

7206	023362	063414	0BUFF		;0BUFF IS BUFF ADDRESS
7207	023364	000	.BYTE	0	;SECTOR 0
7208	023365	000	.BYTE	0	;TRACK 0
7209	023366	000312	312		;CYLINDER 312
7210					
7211	023370	104417	TLOADRK		;LOAD RK REGS
7212	023372	104424	TWAT32		;WAIT FOR INTERRUPT
7213	023374	104002	ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7214					
7215	023376	104422	TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
7216	023400	000000	0		
7217	023402	000004	WCKERR		;WRITE CHECK ERROR
7218	023404	000000	0		
7219	023406	104004	ERROR	4; OR 5,6,7	;REPORT ALL DISCREPANCIES
7220					
7221	023410	104415	SCOPI		;LOCAL LOOP TO 15
7222					

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

7233	023412	023727	001544	063536		CMP	T.BA, #0BUFF+222	;CHECK BA HALT AT PROPER PLACE
7234	023420	001416				BEQ	25	;YES-SKIP
7235	023422	101040				BHI	65	;IF TO HI - SKIP
7236	023424	012737	054021	001450		MOV	#EM11, EM10N	; "INCORRECT BA"
7237	023432	012737	063636	001202		MOV	#0BUFF+222, \$REG10	;GOOD VALUE
7238	023440	013737	001544	001204		MOV	T.BA, \$REG11	;BAD VALUE
7239	023446	012737	050203	060770		MOV	#OPER41, DF010A	; "WRITE CHECK ABORTED WITH WCE"
7240	023454	104010				ERROR	10	
7241								
7242	023456	023727	001542	177511	25:	CMP	T.WC, #-267	;CHECK WORD COUNT AT CORRECT VALUE
7243	023464	001461				BEQ	35	;YES-SKIP
7244	023466	101037				BHI	75	;IF HIGHER - SKIP
7245	023470	012737	053774	001450		MOV	#EM10, EM10N	; "INCORRECT WC"
7246	023476	012737	050203	060770		MOV	#OPER41, DF010A	; "WRITE CHECK ABORTED WITH WCE"
7247	023504	012737	177511	001202		MOV	#-267, \$REG10	;GOOD VALUE
7248	023512	013737	001542	001204		MOV	T.WC, \$REG11	;ERROR VALUE
7249	023520	104010				ERROR	10	
7250	023522	000442				BR	35	;EXIT
7251								
7252	023524	023727	001544	063642	65:	CMP	T.BA, #0BUFF+226	;TEST IF BA AT HI SIDE
7253	023532	101415				BLOS	75	;YES - SKIP
7254	023534	012737	054021	001450		MOV	#EM11, EM10N	;SET MESSAGE
7255	023542	012737	063642	001202		MOV	#0BUFF+226, \$REG10	;GOOD VALUE
7256	023550	013737	001544	001204		MOV	T.BA, \$REG11	;ERROR VALUE
7257	023556	012737	050203	060770		MOV	#OPER41, DF010A	; "WRITE CHECK ABORTED WITH WCE"
7258	023564	104010				ERROR	10	
7259								
7260	023566	023727	001542	177513	75:	CMP	T.WC, #-265	;TEST IF WORD COUNT AT HI SIDE
7261	023574	101415				BLOS	35	;YES - SKIP

H11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 136
T71 WRITE CHECK ERROR (PART 2)

SEQ 0136

7262	023576	012737	053774	001450	MOV	#EM10,EM10M	;SET MESSAGE
7263	023604	012737	050203	060770	MOV	#OPER41,DF010A	; "WC ABORTED WITH WCE"
7264	023612	012737	177513	001202	MOV	#-265,\$REG10	;GOOD VALUE
7265	023620	013737	001542	001204	MOV	T.WC,\$REG11	;ERROR VALUE
7266	023626	104010			ERROR	10	
7267							
7268	023630	104415			3\$: SCOP1		
7269							
7270	023632	032700	100000		BIT	#BIT15,R0	;BIT 15 SET? (ALL PATTERNS TESTED)
7271	023636	001002			BNE	4\$	;YES-EXIT
7272	023640	006300			ASL	R0	;SHIFT FOR NEXT TEST
7273	023642	000637			BR	1\$	;LOOP
7274							
7275	023644				4\$:		
7276					*****		
7277					TEST 72	WRITE CHECK OF PARTIAL SECTOR	
7278					*		
7279					*	WRITE CYLINDER 312, TRACK 0, SECTOR WITH A KNOWN	
7280					*	CONFIGURATIONS. ISSUE A WRITE CHECK COMMAND OF	
7281					*	110 WORDS MAKING SURE THE 111TH WORD IS	
7282					*	DIFFERENT THAN DATA ON DISK. MAKE SURE	
7283					*	WRITE CHECK ERROR DOES NOT SET.	
7284					*		
7285					*****		
7286	023644	000004			↑ST72: SCOPE		
7287	023646	012737	000012	001262	MOV	#10.,\$TIMES	;DO 10. ITERATIONS
7288	023654	104416			TSSINIT		;CLEAR SUBSYSTEM
7289	023656	104003			ERROR	3	;BAD INIT ERROR
7290							
7291	023660	004437	041740		JSR	R4,GENCOM	;GENERATE DATA
7292	023664	000007			7		;ALL 1'S
7293	023666	000400			400		;400 WORDS
7294							
7295	023670	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
7296	023674	000123			WRDATA		;WRDATA
7297	023676	177400			-400		; -400 WORDS
7298	023700	063414			OBUFF		;OBUFF IS BUFF ADDRESS
7299	023702	000			.BYTE	0	;SECTOR 0
7300	023703	000			.BYTE	0	;TRACK 0
7301	023704	000312			312		;CYLINDER 312
7302							
7303	023706	104417			TLOADRK		;LOAD RK REGS
7304	023710	104430			TWAT96		;WAIT FOR INTERRUPT
7305	023712	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7306							
7307	023714	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7308	023716	104004			ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7309							
7310	023720	005037	063636		CLR	OBUFF+222	
7311							
7312	023724	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
7313	023730	000131			WRCHK		;WRCHK
7314	023732	177670			-110		; -110 WORDS
7315	023734	063414			OBUFF		;OBUFF IS BUFF ADDRESS
7316	023736	000			.BYTE	0	;SECTOR 0
7317	023737	000			.BYTE	0	;TRACK 0

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 137
T72 WRITE CHECK OF PARTIAL SECTOR

SEQ 0137

7318	023740	000312		312		;CYLINDER 312
7319						
7320	023742	104417		TLOADRK		;LOAD RK REGS
7321	023744	104424		TWAT32		;WAIT FOR INTERRUPT
7322	023746	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7323						
7324	023750	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7325	023752	104004		ERROR 4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7326						
7327				.SBTTL	**MAXIMUM DATA TRANSFER AND CONTROLLER TIME OUT	
7328						
7329				*****		
7330				*TEST 73	MAXIMUM DATA TRANSFER (PART 1)	
7331				*		
7332				*	IN THE FIRST PASS OF THE PROGRAM, THE HEADERS OF	
7333				*	THE FIRST 4 CYLINDERS ARE WRITTEN. THIS IS DONE TO	
7334				*	INSURE THE FORMAT IS CORRECT.	
7335				*		
7336				*	ZERO OUT THE FIRST 256 SECTORS OF THE DISK WITH	
7337				*	ONE SECTOR WRITES. ISSUE A SEEK TO CYLINDER 0, TRACK 0.	
7338				*	ISSUE A WRITE DATA OF MAXIMUM DATA TRANSFER 200000 WORDS	
7339				*	TO CYLINDER 0, TRACK 0, SECTOR 0. MAKE SURE CONTROLLER	
7340				*	TIME OUT IS NOT SET. CHECK CYLINDER ADDRESS,	
7341				*	DISK ADDRESS, BUS ADDRESS AND WORD COUNT. READ	
7342				*	EACH SECTOR TO MAKE SURE IT WAS WRITTEN CORRECTLY.	
7343				*		
7344				*	NOTE: THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT	
7345				*	IN THE FIRST 256 SECTORS ON THE PACK.	
7346				*		
7347				*	*****	
7348	023754	000004		TST73:	SCOPE	
7349	023756	012737	000002 001262		MOV #2,STIMES	;DO 2 ITERATIONS
7350	023764	032737	000400 001664		BIT #FPFMT,OPTFLG	;TEST IF FIRST PASS SWITCH SET
7351	023772	001043			BNE 24\$	;YES - SKIP FORMAT
7352						
7353	023774	105037	024017		CLRB 21\$	;CLEAR ADDRESS POINTERS
7354	024000	005037	024020		CLR 22\$	
7355						
7356	024004	004437	035530	20\$:	JSR R4,LRLoad	;LOAD "L" REGISTERS
7357	024010	000127			WRHEAD	;WRITE HEADER
7358	024012	177676			-102	;102 WORDS
7359	024014	063414			OBUFF	;OBUFF IS BUFF ADDRESS
7360	024016	000			.BYTE 0	;SECTOR 0
7361	024017	000		21\$:	.BYTE 0	;TRACK ADDRESS (VARIABLE)
7362	024020	000000		22\$:	0	;CYLINDER 0 (VARIABLE)
7363						
7364	024022	004437	041740		JSR R4,GENCOM	;GO GENERATE HEADERS
7365	024026	001200			1200	;WITH BAD SECTOR ERRORS
7366						
7367	024030	104417			TLOADRK	;LOAD RK REGS
7368	024032	104431			TWAT112	;WAIT FOR INTERRUPT
7369	024034	104002			ERROR 2	;TO SLOW/NOT COMPLETE ERROR
7370						
7371	024036	104421			TCHKOP	;CHECK OPERATION FOR ANY ERRORS
7372	024040	104004			ERROR 4	;OR 5, 6, 7 ;REPORT ALL ERRORS
7373						

7374	024042	123727	024017	000002		CMPB	21\$, #2	;TEST IF LAST TRACK
7375	024050	001403				BEG	23\$	;YES - SKIP
7376	024052	105237	024017			INCB	21\$	;ELSE BUMP TRACK
7377	024056	000752				BR	20\$	;LOOP
7378								
7379	024060	105037	024017		23\$:	CLRB	21\$	;CLEAR TRACK POINTER
7380	024064	023727	024020	000003		CMP	22\$, #3	;TEST IF LAST CYLINDER WRITTEN
7381	024072	001403				BEG	24\$	;YES - SKIP
7382	024074	005237	024020			INC	22\$	;ELSE BUMP CYLINDER
7383	024100	000741				BR	20\$	;LOOP
7384								
7385	024102	013737	024114	001110	24\$:	MOV	1\$, \$LPERR	;SET LOCAL LOOP ON ERROR
7386	024110	012703	000400			MOV	#400, R3	;SET COUNT FOR SECTOR CLEARING
7387	024114				1\$:			
7388	024114	104416				TSSINIT		;CLEAR SUBSYSTEM
7389	024116	104003				ERROR	3	;BAD INIT ERROR
7390	024120	004437	035530			JSR	R4, LRLOAD	;LOAD "L" REGS
7391	024124	000123				WRDATA		;WRDATA
7392	024126	177400				-400		; -400 WORDS
7393	024130	063414				OBUFF		;OBUFF IS BUFF ADDRESS
7394	024132	000				.BYTE	0	;SECTOR 0
7395	024133	000				.BYTE	0	;TRACK 0
7396	024134	000000				0		;CYLINDER 0
7397								
7398	024136	104417				TLOADRK		;LOAD RK REGS
7399	024140	104434				TWAIT59		;WAIT FOR INTERRUPT
7400	024142	104002				ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7401								
7402	024144	104421				TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7403	024146	104004				ERROR	4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
7404								
7405	024150	104415				SCOP1		;LOCAL LOOP ON ERROR TO 1\$
7406	024152	005303				DEC	R3	;DECREMENT COUNT
7407	024154	012762	063414	000004	2\$:	MOV	#OBUFF, RKBA(R2)	;SET BA
7408	024162	012762	177400	000002		MOV	#-400, RKWC(R2)	;AND WC AGAIN
7409	024170	005037	001662			CLR	INTSET	;CLEAR INTERRUPT FLAG
7410	024174	013762	001626	000010		MOV	DRVNUM, RKCS2(R2)	;SET DRIVE NUMBER
7411	024202	012762	000123	000000		MOV	#WRDATA, RKCS1(R2)	;DO WRITE DATA
7412								
7413	024212	104425				TWAIT48		;WAIT FOR INTERRUPT
7414	024212	104002				ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7415								
7416	024214	032762	000200	000014		BIT	#BSE, RKER(R2)	;BAD SECTOR ERROR?
7417	024222	001415				BEG	3\$	;NO-SKIP
7418	024224	032737	000200	001664		BIT	#BSERPT, OPTFLG	;TEST IF BSE TO MANY HAS BEEN REPORTED
7419	024232	001007				BNE	5\$	;YES - SKIP
7420	024234	052737	000200	001664		BIS	#BSERPT, OPTFLG	;SET FLAG
7421	024242	012737	052445	001360		MOV	#OPRO17, EMIN	;SET MESSAGE
7422	024250	104001				ERROR	1	; "FIRST 256 SECTOR NOT BSE FREE"
7423	024252	000137	024606		5\$:	JMP	14\$	;GO TO EXIT
7424					3\$:			
7425	024256							
7426	024256	104421				TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7427	024260	104004				ERROR	4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
7428	024262	104415				SCOP1		;LOCAL LOOP TO 1\$ (RESTART SECTOR CLEAR)
7429								

K11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 139
T73 MAXIMUM DATA TRANSFER (PART 1)

SEQ 0139

7430	024264	005303		DEC	R3		; DECREMENT COUNT
7431	024266	001332		BNE	25		; LOOP IF NOT ZERO
7432							
7433	024270	004437	035530	JSR	R4, LRLOAD		; LOAD "L" REGS
7434	024274	000117		SEEK			; SEEK
7435	024276	000000		0			; 0 WORDS
7436	024300	000000		0			; 0 IS BUFF ADDRESS
7437	024302	000		.BYTE	0		; SECTOR 0
7438	024303	000		.BYTE	0		; TRACK 0
7439	024304	000000		0			; CYLINDER 0
7440							
7441	024306	104417		TLOADRK			; LOAD RK REGS
7442	024310	104423		TWAT16			; WAIT FOR INTERRUPT
7443	024312	104002		ERROR	2		; TO SLOW/NOT COMPLETE ERROR
7444	024314	005037	001662	CLR	INTSET		; CLEAR FIRST INTERRUPT
7445	024320	104421		TCHKOP			; CHECK OPERATION FOR ANY ERRORS
7446	024322	104004		ERROR	4 ;OR 5, 6, 7		; REPORT ALL ERRORS
7447	024324	104427		TWAT80			; WAIT FOR SECOND INTERRUPT
7448	024326	104002		ERROR	2		; TO SLOW/NOT COMPLETE ERROR
7449	024330	004437	041740	JSR	R4, GENCOM		; GENERATE DATA
7450	024334	004006		4006			; PATTERN 6, 1ST WORD REPEATED
7451	024336	000400		400			; 400 WORDS
7452							
7453	024340			45:			
7454	024340	104416		TSSINIT			; CLEAR SUBSYSTEM
7455	024342	104003		ERROR	3		; BAD INIT ERROR
7456	024344	004437	035530	JSR	R4, LRLOAD		; LOAD "L" REGS
7457	024350	000123		WRODATA			; WRODATA
7458	024352	000000		0			; 0 WORDS
7459	024354	063414		OBUFF			; OBUFF IS BUFF ADDRESS
7460	024356	000		.BYTE	0		; SECTOR 0
7461	024357	000		.BYTE	0		; TRACK 0
7462	024360	000000		0			; CYLINDER 0
7463							
7464	024362	052737	000020 001610	BIS	#BAI, L.CS2		
7465							
7466	024370	104417		TLOADRK			; LOAD RK REGS
7467	024372	104437		TWAT85			; WAIT FOR SECOND INTERRUPT
7468	024374	104002		ERROR	2		; ELSE REPORT TO SLOW/NOT COMPLETE ERROR
7469							
7470	024376			75:			
7471	024376	104421		TCHKOP			; CHECK OPERATION FOR ANY ERRORS
7472	024400	104004		ERROR	4 ;OR 5, 6, 7, 10		; REPORT ALL ERRORS
7473							
7474	024402	104415		SCOPI			; INTERNAL LOOP ON ERROR TO 45
7475	024404	012703	000400	MOV	#400, R3		; SET COUNTER FOR READ-COMPARE LOOP
7476	024410	005037	024452	CLR	105		; CLEAR SECTOR AND TRACK VALUES
7477	024414	005037	024454	CLR	125		; CLEAR CYL VALUE
7478	024420	013737	024434 001110	MOV	85, \$LPERR		; SET LOCAL LOOP ON ERROR
7479	024426	042737	000020 001610	BIC	#BAI, L.CS2		; CLEAR BAI
7480							
7481	024434			85:			
7482	024434	104416		TSSINIT			; CLEAR SUBSYSTEM
7483	024436	104003		ERROR	3		; BAD INIT ERROR
7484	024440	004437	035530	JSR	R4, LRLOAD		; LOAD RK REGS
7485	024444	000121		RDDATA			; READ DATA

L11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 140
T73 MAXIMUM DATA TRANSFER (PART 1)

SEQ 0140

7486	024446	177400		-400		;400 WORDS
7487	024450	061414		IBUFF		;INTO IBUFF
7488	024452	000	10\$:	.BYTE	0	;SECTOR (VARIABLE)
7489	024453	000	11\$:	.BYTE	0	;TRACK (VARIABLE)
7490	024454	000000	12\$:	0		;CYL (VARIABLE)
7491						
7492	024456	104417		TLOADRK		;LOAD RK REGS
7493	024460	104425		TWAT48		;WAIT FOR INTERRUPT
7494	024462	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
7495						
7496	024464	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7497	024466	104004		ERROR	4 ;OR 5, 6, 7,	10 ;REPORT ALL ERRORS
7498						
7499	024470	104415		SCOP1		;LOCAL LOOP ON ERROR TO B\$
7500						
7501	024472	004437	041740	JSR	R4,GENCOM	;COMPARE DATA
7502	024476	100000		100000		
7503	024500	000400		400		;400 WORDS
7504	024502	000413		BR	13\$	;NO MISCOMPARE-EXIT LOOP
7505	024504	104015		ERROR	15	;REPORT FIRST ERROR
7506						
7507	024506	013700	001634	MOV	ERRLMT,R0	;GET ERROR LIMIT
7508	024512	005300		DEC	R0	;DECREMENT COUNT
7509	024514	001406		BEQ	65\$	;IF ZERO - EXIT
7510	024516	004437	041740	JSR	R4,GENCOM	;CONTINUE DATA COMPARE
7511	024522	040000		40000		
7512	024524	000402		BR	65\$	;NO MORE ERRORS - EXIT
7513	024526	104016		ERROR	16	;REPORT NEXT ERROR
7514	024530	000770		BR	64\$	;LOOP
7515	024532					
7516						
7517	024532	104415		13\$:	SCOP1	;LOCAL LOOP TO B\$
7518						
7519	024534	005303		DEC	R3	;DEC READ LOOP COUNT
7520	024536	001423		BEQ	14\$	;IF ZERO-EXIT
7521						
7522	024540	105237	024452	INCB	10\$	;BUMP SECTOR
7523	024544	123727	024452	CMPB	10\$,#26	;FINISHED WITH TRACK?
7524	024552	001332		BNE	9\$	;NO-LOOP
7525	024554	105037	024452	CLRB	10\$	;CLEAR SECTOR
7526	024560	105237	024453	INCB	11\$	;BUMP TRACK
7527	024564	123727	024453	CMPB	11\$,#3	;FINISHED WITH CYLINDER?
7528	024572	001322		BNE	9\$	;NO-LOOP
7529	024574	105037	024453	CLRB	11\$	;CLEAR TRACK
7530	024600	005237	024454	INC	12\$	;BUMP CYL.
7531	024604	000715		BR	9\$	;LOOP
7532						
7533	024606			14\$:		
7534				*****		
7535				TEST 74	MAXIMUM DATA TRANSFER (PART 2)	
7536				*		
7537				*		
7538				*	ZERO OUT FIRST 256 SECTORS OF THE DISK WITH	
7539				*	200000 WORD WRITE. SEEK TO CYLINDER 632.	
7540				*	ISSUE A WRITE OF MAXIMUM DATA TRANSFER	
7541				*	200000 WORD WRITE. MAKE SURE CONTROLLER TIME	
				*	OUT IS NOT SET. CHECK CYLINDER ADDRESS	

M11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 141
T74 MAXIMUM DATA TRANSFER (PART 2)

SEQ 0141

```

7542      *      DISK ADDRESS, BUS ADDRESS AND WORD COUNT.
7543      *      SEEK TO CYLINDER 632.  ISSUE A WRITE CHECK
7544      *      OF 200000 WORDS.  MAKE SURE NO ERROR SETS.
7545      *
7546      *      NOTE:  THIS TEST IS EXECUTED ONLY IF NO BAD SECTORS ARE PRESENT
7547      *      IN THE FIRST 256 SECTORS ON THE PACK.
7548      *
7549      *      *****
7550 024606 000004      ST74: SCOPE
7551 024610 012737 000002 001262 MOV      #2, $TIMES      ; DO 2 ITERATIONS
7552 024616 104416      TSSINIT      ; CLEAR SUBSYSTEM
7553 024620 104003      ERROR      3      ; BAD INIT ERROR
7554 024622 012700 000620 MOV      #400., R0      ; SET COUNT FOR INTERRUPT WAIT
7555 024626 005037 063414 CLR      OBUFF
7556
7557 024632 004437 035530 JSR      R4, LRLOAD      ; LOAD "L" REGS
7558 024636 000123      WRDATA      ; WRDATA
7559 024640 000000      0      ; 0 WORDS
7560 024642 063414      OBUFF      ; OBUFF IS BUFF ADDRESS
7561 024644      000      .BYTE      0      ; SECTOR 0
7562 024646      000      .BYTE      0      ; TRACK 0
7563 024648 000000      0      ; CYLINDER 0
7564 024650 052737 000020 001610 BIS      #BAI, L.CS2      ; SET BAI
7565 024656 104417      TLOADRK      ; LOAD RK REGS
7566 024660 104434      TWAT159      ; WAIT FOR INTERRUPT
7567 024662 000401      BR      2$      ; NO INTERRUPT-SKIP
7568 024664 000403      BR      3$      ; INTERRUPT-SKIP
7569
7570 024666 005300      2$: DEC      R0      ; DEC WAIT COUNTER
7571 024670 001373      BNE      1$      ; NO ZERO-LOOP
7572 024672 104002      ERROR      2      ; TO SLOW/NOT COMPLETE ERROR
7573
7574 024674 032762 000200 000014 3$: BIT      #BSE, RKER(R2)      ; DID BSE SET
7575 024702 001415      BEQ      4$      ; NO-SKIP
7576
7577 024704 032737 000200 001664 BIT      #BSERPT, OPTFLG      ; TEST IF TO MANY BAD SECTORS REPORTED
7578 024712 001007      BNE      12$      ; YES - SKIP
7579 024714 052737 000200 001664 BIS      #BSERPT, OPTFLG      ; SET FLAG
7580 024722 012737 052445 001360 MOV      #OPR017, EMIN      ; SET MESSAGE
7581 024730 104001      ERROR      1      ; "FIRST 256 SECTORS NOT BSE FREE"
7582 024732 000137 025264      12$: JMP      11$      ; EXIT
7583
7584      4$:
7585 024736      TCHKOP      ; CHECK OPERATION FOR ANY ERRORS
7586 024740 104421      ERROR      4 ;OR 5, 6, 7, 10 ;REPORT ALL ERRORS
7587 104004
7588 024742 004437 035530 JSR      R4, LRLOAD      ; LOAD "L" REGS
7589 024746 000117      SEEK      ; SEEK
7590 024750 000000      0      ; 0 WORDS
7591 024752 000000      0      ; 0 IS BUFF ADDRESS
7592 024754      000      .BYTE      0      ; SECTOR 0
7593 024756      000      .BYTE      0      ; TRACK 0
7594 024758 000632      632      ; CYLINDER 632
7595
7596 024760 104417      TLOADRK      ; LOAD RK REGS
7597 024762 104423      TWAT16      ; WAIT FOR INTERRUPT

```

N11

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 142
T74 MAXIMUM DATA TRANSFER (PART 2)

SEQ 0142

7598	024764	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7599	024766	005037	001662	CLR	INTSET		;CLEAR INTERRUPT FLAG
7600							
7601	024772	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7602	024774	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
7603							
7604	024776	104427		TWAT80			;WAIT FOR 2ND INTERRUPT
7605	025000	104002		ERROR	2		
7606							
7607	025002	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7608	025004	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
7609							
7610	025006	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
7611	025012	000105		CLEAR			;CLEAR
7612	025014	000000		0			;0 WORDS
7613	025016	000000		0			;0 IS BUFF ADDRESS
7614	025020	000		.BYTE	0		;SECTOR 0
7615	025021	000		.BYTE	0		;TRACK 0
7616	025022	000000		0			;CYLINDER 0
7617							
7618	025024	104417		TLOADRK			;LOAD RK REGS
7619	025026	104423		TWAT16			;WAIT FOR INTERRUPT
7620	025030	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7621							
7622	025032	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7623	025034	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
7624							
7625	025036	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
7626	025042	000123		WRDATA			;WRDATA
7627	025044	000000		0			;0 WORDS
7628	025046	063414		0BUFF			;0BUFF IS BUFF ADDRESS
7629	025050	000		.BYTE	0		;SECTOR 0
7630	025051	000		.BYTE	0		;TRACK 0
7631	025052	000000		0			;CYLINDER 0
7632							
7633	025054	012737	135143 063414	MOV	#135143,0BUFF		;SET WORD FOR OUTPUT
7634	025062	012700	000621	MOV	#401,R0		;SET COUNT FOR INTERRUPT WAIT
7635	025066	052737	000020 001610	BIS	#BAI,L.CS2		;SET BUS ADDRESS INC INHIBIT
7636							
7637	025074	104417		TLOADRK			;LOAD RK REGS
7638	025076	104434		TWAT159			;WAIT FOR INTERRUPT
7639	025100	000401		BR	6\$		;NO INTERRUPT-BRANCH
7640	025102	000403		BR	7\$		;INTERRUPT-BRANCH
7641							
7642	025104	005300		DEC	R0		;DEC WAIT COUNT
7643	025106	001373		BNE	5\$		;LOOP IF NOT ZERO
7644	025110	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
7645							
7646	025112						
7647	025112	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
7648	025114	104004		ERROR	4	;OR 5, 6, 7,	;REPORT ALL ERRORS
7649	025116	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
7650	025122	000117		SEEK			;SEEK
7651	025124	000000		0			;0 WORDS
7652	025126	000000		0			;0 IS BUFF ADDRESS
7653	025130	000		.BYTE	0		;SECTOR 0

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 15:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 143
T74 MAXIMUM DATA TRANSFER (PART 2)

SEQ 0143

7654	025131	000		.BYTE 0		;TRACK 0
7655	025132	000632		632		;CYLINDER 632
7656	025134	104417		TLOADRK		;LOAD RK REGS
7657	025136	104423		TWAIT16		;WAIT FOR INTERRUPT
7658	025140	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7659	025142	005037	001662	CLR INTSET		;CLEAR INTERRUPT FLAG
7660						
7661	025146	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7662	025150	104004		ERROR 4	;OR 5, 6, 7	;REPORT ALL ERRORS
7663						
7664	025152	104427		TWAIT80		;WAIT FOR SECOND INIT
7665	025154	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7666	025156	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7667	025160	104004		ERROR 4	;OR 5, 6, 7	;REPORT ALL ERRORS
7668						
7669	025162	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7670	025166	000105		CLEAR		;CLEAR
7671	025170	000000		0		;0 WORDS
7672	025172	000000		0		;0 IS BUFF ADDRESS
7673	025174	000		.BYTE 0		;SECTOR 0
7674	025175	000		.BYTE 0		;TRACK 0
7675	025176	000000		0		;CYLINDER 0
7676						
7677	025200	104417		TLOADRK		;LOAD RK REGS
7678	025202	104423		TWAIT16		;WAIT FOR INTERRUPT
7679	025204	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7680						
7681	025206	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7682	025210	104004		ERROR 4	;OR 5, 6, 7	;REPORT ALL ERRORS
7683						
7684	025212	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7685	025216	000131		WRTCHK		;WRTCHK
7686	025220	000000		0		;0 WORDS
7687	025222	063414		OBUFF		;OBUFF IS BUFF ADDRESS
7688	025224	000		.BYTE 0		;SECTOR 0
7689	025225	000		.BYTE 0		;TRACK 0
7690	025226	000000		0		;CYLINDER 0
7691	025230	052737	000020 001610	BIS #BAI,L.CS2		;SET BAI FLAG
7692	025236	012700	000621	MOV #401.,R0		;SET WAIT COUNT
7693						
7694	025242	104417		TLOADRK		;LOAD RK REGS
7695	025244	104434		TWAIT159		;WAIT FOR INTERRUPT
7696	025246	000401		BR 95		;NO INTERRUPT-SKIP
7697	025250	000403		BR 105		;INTERRUPT-SKIP
7698						
7699	025252	005300		DEC R0		;DEC WAIT COUNT
7700	025254	001373		BNE 85		;NOT ZERO-LOOP
7701	025256	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7702						
7703	025260					
7704	025260	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7705	025262	104004		ERROR 4	;OR 5, 6, 7, 10	;REPORT ALL ERRORS
7706						
7707	025264					
7708						
7709						

85: ;*****
95: ;TEST 75 CONTROLLER TIME OUT
105: ;*****

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 144
T75 CONTROLLER TIME OUT

SEQ 0144

```

7710
7711
7712
7713
7714
7715
7716
7717 025264 000004
7718 025266 012737 000005 001262
7719 025274 104416
7720 025276 104003
7721
7722 025300 004437 035530
7723 025304 000117
7724 025306 000000
7725 025310 000000
7726 025312 000
7727 025313 000
7728 025314 000632
7729
7730 025316 104417
7731 025320 104423
7732 025322 104002
7733
7734 025324 104421
7735 025326 104004
7736
7737 025330 005037 001662
7738 025334 104427
7739 025336 104002
7740 025340 104421
7741 025342 104004
7742
7743 025344 004437 035530
7744 025350 000105
7745 025352 000000
7746 025354 000000
7747 025356 000
7748 025357 000
7749 025360 000000
7750
7751 025362 104417
7752 025364 104423
7753 025366 104002
7754
7755 025370 104421
7756 025372 104004
7757
7758 025374 004437 035530
7759 025400 000113
7760 025402 000000
7761 025404 000000
7762 025406 000
7763 025407 000
7764 025410 000000
7765

; *
; * 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.
; *
; *****
; ST75: SCOPE
; MOV #5, $TIMES ; DO 5. ITERATIONS
; 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
;
; TLOADRK ; LOAD RK REGS
; TWP*16 ; WAIT FOR INTERRUPT
; ERROR 2 ; TO SLOW/NOT COMPLETE ERROR
;
; TCHKOP ; CHECK OPERATION FOR ANY ERRORS
; ERROR 4 ;OR 5, 6, 7 ; REPORT ALL ERRORS
;
; CLR INTSET ; CLEAR INTERRUPT FLAG
; TWP*80 ; WAIT FOR SECOND INTERRUPT
; ERROR 2 ; TO SLOW/NOT COMPLETE ERROR
; TCHKOP ; CHECK OPERATION FOR ANY ERRORS
; ERROR 4 ;OR 5, 6, 7 ; REPORT ALL ERRORS
;
; JSR R4, LRLOAD ; LOAD "L" REGS
; CLEAR ; CLEAR
; 0 ; 0 WORDS
; 0 ; 0 IS BUFF ADDRESS
; .BYTE 0 ; SECTOR 0
; .BYTE 0 ; TRACK 0
; 0 ; CYLINDER 0
;
; TLOADRK ; LOAD RK REGS
; TWP*16 ; 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
;
; 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

```

7766	025412	104417		TLOADRK		;LOAD RK REGS
7767	025414	104423		TWAT16		;WAIT FOR INTERRUPT
7768	025416	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7769						
7770	025420	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7771	025424	000125		RDHEAD		;RDHEAD
7772	025426	000000		0		;0 WORDS
7773	025430	000000		0		;0 IS BUFF ADDRESS
7774	025432	000		.BYTE 0		;SECTOR 0
7775	025433	000		.BYTE 0		;TRACK 0
7776	025434	000000		0		;CYLINDER 0
7777						
7778	025436	104417		TLOADRK		;LOAD RK REGS
7779	025440	104436		TWAT25		;WAIT FOR INTERRUPT
7780	025442	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7781						
7782	025444	104422		TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
7783	025446	000000		0		
7784	025450	000000		0		
7785	025452	000002		CTOERR		;CONTROLLER TIME OUT
7786	025454	104004		ERROR 4; OR 5,6,7		;REPORT ANY DISCREPANCIES
7787	025456	104416		TSSINIT		;CLEAR SUBSYSTEM
7788	025460	104003		ERROR 3		;BAD INIT ERROR
7789	025462	005037	001662	CLR INTSET		;CLEAR INT FLAG
7790	025466	012762	000100 000000	MOV #IE,RKCS1(R2)		;SET INT ENABLE
7791	025474	104437		TWAT8S		;WAIT FOR SECOND INT
7792	025476	104002		ERROR 2		
7793						
7794						
7795						
7796						
7797						
7798						
7799						
7800						
7801						
7802						
7803						
7804						
7805						
7806						
7807						
7808	025500	000004		TEST 76: SCOPE		
7809	025502	012737	000003 001262	MOV #3,\$TIMES		;DO 3. ITERATIONS
7810						
7811	025510	104416		TSSINIT		;CLEAR SUBSYSTEM
7812	025512	104003		ERROR 3		;BAD INIT ERROR
7813						
7814	025514	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7815	025520	000113		RECAL		;RECAL
7816	025522	000000		0		;0 WORDS
7817	025524	000000		0		;0 IS BUFF ADDRESS
7818	025526	000		.BYTE 0		;SECTOR 0
7819	025527	000		.BYTE 0		;TRACK 0
7820	025530	000000		0		;CYLINDER 0
7821						

.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.
*****

```

```

;*****
;ST76:  SCOPE
;      MOV      #3,$TIMES      ;;DO 3. ITERATIONS
;
;      TSSINIT
;      ERROR    3      ;CLEAR SUBSYSTEM
;                      ;BAD INIT ERROR
;
;      JSR      R4,LRLOAD      ;LOAD "L" REGS
;      RECAL
;      0
;      0
;      0
;      .BYTE    0      ;RECAL
;                      ;0 WORDS
;                      ;0 IS BUFF ADDRESS
;      .BYTE    0      ;SECTOR 0
;                      ;TRACK 0
;      0
;                      ;CYLINDER 0
;*****

```

E12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 146
 02R6KD.P11 31-JAN-77 16:26 T76 LIMIT DETECT ON DATA TRANSFER

SEQ 0146

7822	025532	104417		TLOADRK		;LOAD RK REGS
7823	025534	104423		TWAT16		;WAIT FOR INTERRUPT
7824	025536	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7825						
7826	025540	005037	001662	CLR INTSET		;CLEAR INTERRUPT FLAG
7827	025544	104437		TWAT85		;WAIT FOR SECOND INTERRUPT
7828	025546	104002		ERROR 2		
7829						
7830	025550	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
7831	025552	104004		ERROR 4 ;OR 5, 6, 7		;REPORT ALL ERRORS
7832						
7833	025554	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7834	025560	000117		SEEK		;SEEK
7835	025562	000000		0		;0 WORDS
7836	025564	000000		0		;0 IS BUFF ADDRESS
7837	025566	000		.BYTE 0		;SECTOR 0
7838	025567	000		.BYTE 0		;TRACK 0
7839	025570	000002		2		;CYLINDER 2
7840	025572	012737	000020 001615	MOV #PAT,L.MR1		;SET EVEN PARITY BIT
7841	025600	104417		TLOADRK		;LOAD RK REGS
7842	025602	104423		TWAT16		;WAIT FOR INTERRUPT
7843	025604	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7844						
7845	025606	104416		TSSINIT		;CLEAR SUBSYSTEM
7846	025610	104003		ERROR 3		;BAD INIT ERROR
7847						
7848	025612	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7849	025616	000123		WRDATA		;WRDATA
7850	025620	177400		-400		; -400 WORDS
7851	025622	063414		OBUFF		;OBUFF IS BUFF ADDRESS
7852	025624	000		.BYTE 0		;SECTOR 0
7853	025625	000		.BYTE 0		;TRACK 0
7854	025626	000001		1		;CYLINDER 1
7855						
7856	025630	104417		TLOADRK		;LOAD RK REGS
7857	025632	104423		TWAT16		;WAIT FOR INTERRUPT
7858	025634	104002		ERROR 2		;TO SLOW/NOT COMPLETE ERROR
7859						
7860	025636	104422		TCHKWE		;CHECK OPERATION WITH ERROR
7861	025640	000002		SKIERR		;SEEK INCOMPLETE
7862	025642	000000		0		
7863	025644	000000		0		
7864	025646	104004		ERROR 4 ;OR 5,6,OR7		;REPORT ALL DISCREPANCIES
7865						
7866	025650	104416		TSSINIT		;CLEAR SUBSYSTEM
7867	025652	104003		ERROR 3		;BAD INIT ERROR
7868						
7869						
7870	025654					
7871	025654	004437	035530	JSR R4,LRLOAD		;LOAD "L" REGS
7872	025660	000101		SELDIV		;SELDIV
7873	025662	000000		0		;0 WORDS
7874	025664	000000		0		;0 IS BUFF ADDRESS
7875	025666	000		.BYTE 0		;SECTOR 0
7876	025667	000		.BYTE 0		;TRACK 0
7877	025670	000000		0		;CYLINDER 0

35:

F12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 147
T76 LIMIT DETECT ON DATA TRANSFER

SEQ 0147

```

7878
7879 025672 012737 000001 001616      MOV    #1,L.MR1      ;SET TO GET STATUS PAIR 1
7880 025700 104417                    TLOADRK             ;LOAD RK REGS
7881 025702 104423                    TWAT16             ;WAIT FOR INTERRUPT
7882 025704 104002                    ERROR    2          ;TO SLOW/NOT COMPLETE ERROR
7883
7884 025706 104421                    TCHKOP             ;CHECK OPERATION FOR ANY ERRORS
7885 025710 104004                    ERROR    4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
7886
7887 025712 032737 000040 001574      BIT     #5,H0HM,T.MR2 ;TEST IF HEADS HOME
7888 025720 001755                    BEQ      35          ;NO - GET STATUS AGAIN
7889
7890 025722 104416                    TSSINIT            ;CLEAR SUBSYSTEM
7891 025724 104003                    ERROR    3          ;BAD INIT ERROR
7892
7893 025726 012762 000100 000000      MOV     #IE,RKCS1(R2) ;SET IE
7894
7895 025734 005037 001662              CLR     INTSET        ;CLEAR INT FLAG
7896 025740 104437                    TWATBS             ;WAIT FOR SECOND INTERRUPT
7897 025742 000401                    BR       15
7898 025744 000404                    BR       25
7899
7900 025746 012737 060302 001372 15:  MOV     #DH016,DH2N    ;"SUBSYSTEM CLEAR TO RESET LIMIT ERROR
7901                                     ERROR    2          ;ALLOWING HEADS TO RELOAD"
7902
7903
7904 025756                                     25:
7905
7906                                     ;*****
7907                                     ;TEST 77      PROGRAMMING ERROR
7908                                     ;
7909                                     ;ISSUE A SUBSYSTEM CLEAR. ISSUE
7910                                     ;A READ DATA OF 400 WORDS ON CYLINDER 0,
7911                                     ;TRACK 0, SECTOR 0. DURING READ ISSUE A
7912                                     ;WRITE TO THE SPARE REGISTER. MAKE SURE
7913                                     ;PROGRAMMING ERROR SETS.
7914                                     ;*****
7915 ST77: SCOPE
7916 025756 000004 000012 001262      MOV     #10.,$TIMES    ;DO 10. ITERATIONS
7917 025760 012737                    TSSINIT            ;CLEAR SUBSYSTEM
7918 025766 104416                    ERROR    3          ;BAD INIT ERROR
7919 025770 104003
7920 025772 004437 035530              JSR     R4,LRLoad      ;LOAD "L" REGS
7921 025776 000121                    RDATA             ;RDATA
7922 026000 177400                    -400             ;-400 WORDS
7923 026002 061414                    IBUFF            ;IBUFF IS BUFF ADDRESS
7924 026004 000             .BYTE    0          ;SECTOR 0
7925 026005 000             .BYTE    0          ;TRACK 0
7926 026006 000000                    0             ;CYLINDER 0
7927 026010 104417                    TLOADRK             ;LOAD RK REGS
7928
7929 026012 012762 000001 000022      MOV     #1,RKSPAR(R2) ;WRITE SPARE REGISTER
7930
7931 026020 104423                    TWAT16             ;WAIT FOR INTERRUPT
7932 026022 104002                    ERROR    2          ;TO SLOW/NOT COMPLETE ERROR
7933

```

G12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 148
T77 PROGRAMMING ERROR

SEQ 0148

7934	026024	104422		TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
7935	026026	000000		0		
7936	026030	000000		0		
7937	026032	000020		PGERR		;PROG ERROR
7938	026034	104004		ERROR 4	;OR 5,6,7	;REPORT ALL DISCREPANCIES
7939						
7940						
7941						
7942						
7943						
7944						
7945						
7946						
7947						
7948						
7949						
7950						
7951						
7952	026036	000004				
7953	026040	012737	000012 001262			
7954	026046	104416				
7955	026050	104003				
7956						
7957	026052	004437	041740			
7958	026056	000001				
7959	026060	000400				
7960						
7961	026062	004437	035530			
7962	026066	000123				
7963	026070	177400				
7964	026072	063414				
7965	026074	000				
7966	026075	000				
7967	026076	000000				
7968						
7969	026100	104417				
7970	026102	104430				
7971	026104	104002				
7972						
7973	026106	104421				
7974	026110	104004				
7975						
7976	026112	004437	041740			
7977	026116	000002				
7978	026120	000400				
7979						
7980	026122	004437	035530			
7981	026126	000123				
7982	026130	177630				
7983	026132	063414				
7984	026134	000				
7985	026135	000				
7986	026136	000000				
7987						
7988	026140	104417				
7989						


```

*****
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.
*****
TST100: SCOPE
MOV #10., $TIMES ;DO 10. ITERATIONS
TSSINIT ;CLEAR SUBSYSTEM
ERROR 3 ;BAD INIT ERROR

JSR R4, GENCOM ;GENERATE DATA OF ALL ONES
1
400

JSR R4, LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-400 ;-400 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0

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 ;GENERATE DATA OF ZEROS
2
400

JSR R4, LRLOAD ;LOAD "L" REGS
WRDATA ;WRDATA
-150 ;-150 WORDS
OBUFF ;OBUFF IS BUFF ADDRESS
.BYTE 0 ;SECTOR 0
.BYTE 0 ;TRACK 0
0 ;CYLINDER 0

TLOADRK ;START OPERATION

```

SEQ 0149

Address	Hex	Dec	Hex	Dec	Hex	Dec	Instruction	Comment
7990	026142	005737	001662				TST INTSET	;CHECK IF INTERRUPT HAS OCCURRED
7991	026146	001026					BNE 2\$	;YES - MUCH TO SOON. REPORT ERROR
7992	026150	005762	000002				TST RKWC(R2)	;TEST IF NPR'S DONE
7993	026154	001372					BNE 1\$	;NO - LOOP
7994								
7995	026156	052762	100000	000000			BIS #CCLR,RKCS1(R2)	;CLEAR CONTROLLER (CROWBAR WRITE)
7996								
7997	026164	004437	035530				JSR R4,LRLOAD	;LOAD "L" REGS
7998	026170	000121					RDDATA	;RDDATA
7999	026172	177400					-400	; -400 WORDS
8000	026174	061414					IBUFF	;IBUFF IS BUFF ADDRESS
8001	026176	000					.BYTE 0	;SECTOR 0
8002	026177	000					.BYTE 0	;TRACK 0
8003	026200	000000					0	;CYLINDER 0
8004								
8005	026202	104417					TLOADRK	;LOAD RK REGS
8006	026204	104425					TWAIT48	;WAIT FOR INTERRUPT
8007	026206	104002					ERROR 2	;TO SLOW/NOT COMPLETE ERROR
8008								
8009	026210	104422					TCHKWE	;CHECK OPERATION WITH ERROR
8010	026212	000000					0	
8011	026214	000003					DCKERR!ECHERR	;DATA CHECK AND ECC HARD
8012	026216	000000					0	
8013	026220	104004					ERROR 4 ;OR 5,6,7	;REPORT ALL DISCREPANCIES
8014								
8015	026222	000402					BR 3\$	;SKIP TO EXIT
8016	026224							
8017	026224	104421					TCHKOP	;CHECK OPERATION FOR ANY ERRORS
8018	026226	104004					ERROR 4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
8019	026230							
8020								
8021								
8022								
8023								
8024								
8025								
8026								
8027								
8028								
8029								
8030								
8031								
8032	026230	000004					TST101: SCOPE	
8033	026232	012737	000005	001262			MOV #5.,\$TIMES	;DO 5. ITERATIONS
8034								
8035	026240	104416					TSSINIT	;CLEAR SUBSYSTEM
8036	026242	104003					ERROR 3	;BAD INIT ERROR
8037								
8038	026244	004437	035530				JSR R4,LRLOAD	;LOAD "L" REGS
8039	026250	000117					SEEK	;SEEK
8040	026252	000000					0	;0 WORDS
8041	026254	000000					0	;0 IS BUFF ADDRESS
8042	026256	000					.BYTE 0	;SECTOR 0
8043	026257	000					.BYTE 0	;TRACK 0
8044	026260	000632					632	;CYLINDER 63

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 150
T101 DRIVE TIMING ERROR

SEQ 0150

8046	026262	104417			TLOADRK		;LOAD RK REGS
8047	026264	104423			TWAT16		;WAIT FOR INTERRUPT
8048	026266	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8049	026270	005037	001662		CLR	INTSET	;CLEAR INT FLAG
8050	026274	104430			TWAT96		;WAIT FOR SECOND INTERRUPT
8051	026276	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8052							
8053	026300	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
8054	026304	000113			RECAL		;RECAL
8055	026306	000000			0		;0 WORDS
8056	026310	000000			0		;0 IS BUFF ADDRESS
8057	026312	000			.BYTE	0	;SECTOR 0
8058	026313	000			.BYTE	0	;TRACK 0
8059	026314	000000			0		;CYLINDER 0
8060							
8061	026316	104417			TLOADRK		;LOAD RK REGS
8062	026320	104423			TWAT16		;WAIT FOR INTERRUPT
8063	026322	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8064							
8065	026324	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
8066	026330	000125			RDHEAD		;RDHEAD
8067	026332	000000			0		;0 WORDS
8068	026334	000000			0		;0 IS BUFF ADDRESS
8069	026336	000			.BYTE	0	;SECTOR 0
8070	026337	000			.BYTE	0	;TRACK 0
8071	026340	000000			0		;CYLINDER 0
8072	026342	012737	000040	001616	MOV	#DMD,L.MR1	;SET DIAG MODE
8073	026350	104417			TLOADRK		;LOAD RK REGS
8074							
8075	026352	004437	036004		JSR	R4,MCLOCK	;CLOCK CONTROLLER THROUGH SEEK
8076	026356	001062			1062		;AND CLEAR TO READ
8077							
8078	026360	005062	000026		CLR	RKMR1(R2)	;RESET DIAG MODE, LET RD HDRS COMPLETE
8079							
8080	026364	104424			TWAT32		;WAIT FOR INTERRUPT
8081	026366	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8082	026370	104422			TCHKWE		;CHECK OPERATION WITH EXP ERROR
8083	026372	010000			DTERR		;DRIVE TIMING ERROR
8084	026374	000000			0		
8085	026376	000000			0		
8086	026400	104004			ERROR	4 ;OR 5,6,7	;REPORT ALL DISCREPANCIES
8087							
8088	026402						
8089	026402	104416			TSSINIT		;CLEAR SUBSYSTEM
8090	026404	104003			ERROR	3	;BAD INIT ERROR
8091	026406	012762	000100	000000	MOV	#IE,RKCS1(R2)	;SET INTERRUPT ENABLE
8092	026414	005037	001662		CLR	INTSET	;CLEAR INT FLAG
8093							
8094	026420	104437			TWAT85		;WAIT FOR INTERRUPT FOR END OF RECAL
8095	026422	104002			ERROR	2	
8096							
8097	026424	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
8098	026430	000105			CLEAR		;CLEAR
8099	026432	000000			0		;0 WORDS
8100	026434	000000			0		;0 IS BUFF ADDRESS
8101	026436	000			.BYTE	0	;SECTOR 0

15:

J12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 151
T101 DRIVE TIMING ERROR

SEQ 0151

```

8102 026437 000 .BYTE 0 ;TRACK 0
8103 026440 000000 0 ;CYLINDER 0
8104
8105 026442 104417 TLOADRK ;LOAD RK REGS
8106 026444 104423 TWAT16 ;WAIT FOR INTERRUPT
8107 026446 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8108 026450 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8109 026452 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8110
8111 .SBTTL **ERROR FORCING IN DRIVE
8112
8113 *****
8114 *TEST 102 INITIALIZE CLEARING SACK
8115 *
8116 * ISSUE A SUBSYSTEM CLEAR. SELECT AN AVAILABLE
8117 * DRIVE. ISSUE A SUBSYSTEM CLEAR. PUT CONTROLLER IN
8118 * DIAGNOSTIC MODE. ISSUE A SELECT COMMAND WITH
8119 * MESSAGE ID = 3 AND DRIVE SELECTED = 0. CLOCK THROUGH
8120 * PHASE ADDRESS 6. TURN OFF DIAGNOSTIC MODE. MAKE
8121 * SURE UNIT FIELD ERROR DOES NOT SET.
8122 *
8123 *****
8124 026454 000004 ST102: SCCPE
8125 026456 012737 000012 001262 MOV #10., $TIMES ;DO 10. ITERATIONS
8126 026464 104416 TSSINIT ;CLEAR SUBSYSTEM
8127 026466 104003 ERROR 3 ;BAD INIT ERROR
8128
8129 026470 004437 035530 JSR R4, LRLOAD ;LOAD "L" REGS
8130 026474 000101 SELDRV ;SELDV
8131 026476 000000 0 ;0 WORDS
8132 026500 000000 0 ;0 IS BUFF ADDRESS
8133 026502 000 .BYTE 0 ;SECTOR 0
8134 026503 000 .BYTE 0 ;TRACK 0
8135 026504 000000 0 ;CYLINDER 0
8136
8137 026506 104417 TLOADRK ;LOAD RK REGS
8138 026510 104423 TWAT16 ;WAIT FOR INTERRUPT
8139 026512 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8140
8141 026514 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8142 026516 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8143
8144 026520 104416 TSSINIT ;CLEAR SUBSYSTEM
8145 026522 104003 ERROR 3 ;BAD INIT ERROR
8146
8147 026524 004437 035530 JSR R4, LRLOAD ;LOAD "L" REGS
8148 026530 000101 SELDRV ;SELDV
8149 026532 000000 0 ;0 WORDS
8150 026534 000000 0 ;0 IS BUFF ADDRESS
8151 026536 000 .BYTE 0 ;SECTOR 0
8152 026537 000 .BYTE 0 ;TRACK 0
8153 026540 000000 0 ;CYLINDER 0
8154 026542 012737 000043 001616 MOV #3!DMD, L.MR1 ;SET DIAG MODE AND MESSAGE PAIR 3
8155 026550 005037 001610 CLR L.CS2 ;SELECT DRIVE 0
8156
8157 026554 104417 TLOADRK ;LOAD RK REGS

```

K12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 152
T102 INITIALIZE CLEARING SACK

SEQ 0152

```

8158
8159 026556 004437 036004 JSR R4,MCLOCK ;CLOCK THROUGH PHASE ADDRESS 6
8160 026562 001027 1027
8161
8162 026564 042762 000040 000026 BIC #DMD,RKMRI(R2) ;CLEAR MAINTENANCE MODE
8163
8164 026572 104424 TWAT32 ;WAIT FOR INTERRUPT
8165 026574 104002 ERROR 2 ;TO SLOW/NOT COMPLETE
8166
8167 026576 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS
8168 026600 104004 ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8169
8170 *****
8171 :TEST 103 DRIVE OFF TRACK
8172
8173 :
8174 : ISSUE A SUBSYSTEM CLEAR. ISSUE A RECALIBRATE. ISSUE
8175 : OFFSET OF +1200 MICRO-INCHES. PUT CONTROLLER IN DIAGNOSTIC
8176 : MODE. ISSUE A WRITE DATA OF 1 WORD TO CYLINDER 0,
8177 : TRACK 0, SECTOR 0. CLOCK THROUGH SEEK AND DRIVE CLEAR
8178 : MESSAGES. TURN OFF DIAGNOSTIC MODE. DRIVE OFF TRACK
8179 : SHOULD SET IN DRIVE. REPEAT FOR ALL AVAILABLE DRIVES.
8180 *****
8181 :ST103: SCOPE
8182 MOV #5, $TIMES ;DO 5. ITERATIONS
8183 TSSINIT ;CLEAR SUBSYSTEM
8184 ERROR 3 ;BAD INIT ERROR
8185
8186 026602 000004 JSR R4,LRLOAD ;LOAD "L" REGS
8187 026604 012737 000005 001262 RECAL ;RECAL
8188 026612 104416 0 ;0 WORDS
8189 026614 104003 0 ;0 IS BUFF ADDRESS
8190
8191 026616 004437 035530 .BYTE 0 ;SECTOR 0
8192 026622 000113 .BYTE 0 ;TRACK 0
8193 026624 000000 0 ;CYLINDER 0
8194 026626 000000
8195 026630 000 .TLOADRK ;LOAD RK REGS
8196 026631 000 TWAT16 ;WAIT FOR INTERRUPT
8197 026632 000000 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8198
8199 026634 104417 CLR INTSET ;CLEAR INTERRUPT FLAG
8200 026636 104423
8201 026640 104002 TWATBS ;WAIT FOR INTERRUPT #2
8202 026642 005037 001662 ERROR 2
8203
8204 026646 104437 JSR R4,LRLOAD ;LOAD "L" REGS
8205 026648 104002 CLEAR ;CLEAR
8206 026650 104002 0 ;0 WORDS
8207 026652 004437 035530 0 ;0 IS BUFF ADDRESS
8208 026654 000105 .BYTE 0 ;SECTOR 0
8209 026656 000000 .BYTE 0 ;TRACK 0
8210 026658 000000 0 ;CYLINDER 0
8211 026660 000000
8212 026662 000000
8213 026664 000000
8214 026666 000000
8215 026668 000000
8216 026670 104417 .TLOADRK ;LOAD RK REGS
8217 026672 104423 TWAT16 ;WAIT FOR INTERRUPT
8218 026674 104002 ERROR 2 ;TO SLOW/NOT COMPLETE ERROR
8219
8220 026676 104421 TCHKOP ;CHECK OPERATION FOR ANY ERRORS

```

L12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 153
T103 DRIVE OFF TRACK

SEQ 0153

8214	026700	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8215						
8216	026702	004437	035530	JSR	R4,LRLOAD	;LOAD "L" REGS
8217	026706	000115		OFFSET		;OFFSET
8218	026710	000000		0		;0 WORDS
8219	026712	000000		0		;0 IS BUFF ADDRESS
8220	026714	000		.BYTE	0	;SECTOR 0
8221	026715	000		.BYTE	0	;TRACK 0
8222	026716	000000		0		;CYLINDER 0
8223	026720	112737	000060 001612	MOVB	#60,L.ASOF	;SET OFFSET AT +1200
8224						
8225	026726	104417		TLOADRK		;LOAD RK REGS
8226	026730	104423		TWAT16		;WAIT FOR INTERRUPT
8227	026732	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8228						
8229	026734	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8230	026736	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8231						
8232	026740	005037	001662	CLR	INTSET	;CLEAR INT FLAG
8233						
8234	026744	104424		TWAT32		;WAIT FOR INT #2
8235	026746	104002		ERROR	2	
8236						
8237	026750	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8238	026752	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8239						
8240	026754	004437	035530	JSR	R4,LRLOAD	;LOAD "L" REGS
8241	026760	000105		CLEAR		;CLEAR
8242	026762	000000		0		;0 WORDS
8243	026764	000000		0		;0 IS BUFF ADDRESS
8244	026766	000		.BYTE	0	;SECTOR 0
8245	026767	000		.BYTE	0	;TRACK 0
8246	026770	000000		0		;CYLINDER 0
8247						
8248	026772	104417		TLOADRK		;LOAD RK REGS
8249	026774	104423		TWAT16		;WAIT FOR INTERRUPT
8250	026776	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8251						
8252	027000	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8253	027002	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8254						
8255	027004	004437	035530	JSR	R4,LRLOAD	;LOAD "L" REGS
8256	027010	000023		23		;WRITE DATA W/O INTERRUPT ENABLE
8257	027012	177777		-1		; -1 WORDS
8258	027014	063414		0BUFF		;0BUFF IS BUFF ADDRESS
8259	027016	000		.BYTE	0	;SECTOR 0
8260	027017	000		.BYTE	0	;TRACK 0
8261	027020	000000		0		;CYLINDER 0
8262	027022	012737	000040 001616	MOV	#DMD,L.MR1	;SET DIAGNOSTIC MODE
8263						
8264	027030	104417		TLOADRK		
8265						
8266	027032	004437	036004	JSR	R4,MCLOCK	;CLOCK THROUGH SEEK & DRIVE CLEAR
8267	027036	001064		1064		
8268						
8269	027040	005062	000026	CLR	RKMR1(R2)	;CLEAR DIAGNOSTIC MODE

M12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 154
T103 DRIVE OFF TRACK

SEQ 0154

8270	027044	104426			TWAT64		;WAIT FOR INTERRUPT
8271	027046	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR CONTROLLER
8272	027054	104423			TWAT16		;STALL FOR 16 MS
8273	027056	000240			NOP		
8274	027060	000240			NOP		
8275							
8276	027062	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
8277	027066	000101			SELDV		;SELDV
8278	027070	000000			0		;0 WORDS
8279	027072	000000			0		;0 IS BUFF ADDRESS
8280	027074	000			.BYTE	0	;SECTOR 0
8281	027075	000			.BYTE	0	;TRACK 0
8282	027076	000000			0		;CYLINDER 0
8283	027100	005037	001616		CLR	L.MR1	;RESET DIAG MODE
8284							
8285	027104	104417			TLOADRK		;LOAD RK REGS
8286	027106	104423			TWAT16		;WAIT FOR INTERRUPT
8287	027110	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8288							
8289	027112	104422			TCHKWE		;CHECK OPERATION WITH ERROR EXPECTED
8290	027114	000400			DROTERR		;DRIVE OFF TRACK
8291	027116	000000			0		
8292	027120	000000			0		
8293	027122	104004			ERROR	4; OR 5,6,7	;REPORT ANY DISCREPANCIES
8294					*****		
8295					*TEST 104 FILE UNSAFE		
8296					*		
8297					* ISSUE A SUBSYSTEM CLEAR. ISSUE A RECLAIBRATE. ISSUE		
8298					* A READ HEAD OF CYLINDER 0, TRACK 0 IN 24 SECTOR		
8299					* FORMAT. DO A SELECT COMMAND IN 26 SECTOR FORMAT.		
8300					* PUT CONTROLLER IN DIAGNOSTIC MODE. ISSUE A WRITE		
8301					* HEADER TO CYLINDER 0, TRACK 0, ONE WORD IN 26 SECTOR		
8302					* FORMAT. CLOCK THROUGH SEEK AND DRIVE CLEAR MESSAGES.		
8303					* SIMULATE INDEX PULSE. TURN OFF DIAGNOSTIC MODE. FILE		
8304					* UNSAFE SHOULD SET BECAUSE OF ATTEMPTING TO WRITE		
8305					* THROUGH SECTOR PULSE. REPEAT FOR ALL AVAILABLE DRIVES.		
8306					*		
8307					*****		
8308	027124	000004			*ST104: SCOPE		
8309	027126	012737	000005	001262	MOV	#5, \$TIMES	;DO 5. ITERATIONS
8310	027134	012737	177777	001676	MOV	#-1,REFMT	;SET REFORMAT SWITCH
8311	027142	104416			TSSINIT		;CLEAR SUBSYSTEM
8312	027144	104003			ERROR	3	;BAD INIT ERROR
8313							
8314	027146	004437	035530		JSR	R4,LRLOAD	;LOAD "L" REGS
8315	027152	000113			RECAL		;RECAL
8316	027154	000000			0		;0 WORDS
8317	027156	000000			0		;0 IS BUFF ADDRESS
8318	027160	000			.BYTE	0	;SECTOR 0
8319	027161	000			.BYTE	0	;TRACK 0
8320	027162	000000			0		;CYLINDER 0
8321							
8322	027164	104417			TLOADRK		;LOAD RK REGS
8323	027166	104423			TWAT16		;WAIT FOR INTERRUPT
8324	027170	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8325							

N12

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 155
 DZR6KD.P11 31-JAN-77 16:26 T104 FILE UNSAFE

SEQ 0155

8326	027172	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8327	027174	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
8328							
8329	027176	005037	001662	CLR	INTSET		;CLEAR INT FLAG
8330	027202	104437		TWAT8S			;WAIT FOR SECOND INT
8331	027204	104002		ERROR	2		
8332							
8333	027206	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8334	027210	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
8335							
8336	027212	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
8337	027216	000105		CLEAR			;CLEAR
8338	027220	000000		0			;0 WORDS
8339	027222	000000		0			;0 IS BUFF ADDRESS
8340	027224	000		.BYTE	0		;SECTOR 0
8341	027225	000		.BYTE	0		;TRACK 0
8342	027226	000000		0			;CYLINDER 0
8343							
8344	027230	104417		TLOADRK			;LOAD RK REGS
8345	027232	104423		TWAT16			;WAIT FOR INTERRUPT
8346	027234	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8347							
8348	027236	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8349	027240	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
8350							
8351	027242	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
8352	027246	010125		RDHEAD!CFMT			;RDHEAD!CFMT
8353	027250	000000		0			;0 WORDS
8354	027252	000000		0			;0 IS BUFF ADDRESS
8355	027254	000		.BYTE	0		;SECTOR 0
8356	027255	000		.BYTE	0		;TRACK 0
8357	027256	000000		0			;CYLINDER 0
8358							
8359	027260	104417		TLOADRK			;LOAD RK REGS
8360	027262	104424		TWAT32			;WAIT FOR INTERRUPT
8361	027264	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8362							
8363	027266	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
8364	027270	104004		ERROR	4	;OR 5, 6, 7	;REPORT ALL ERRORS
8365							
8366	027272	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
8367	027276	000101		SELDV			;SELDV
8368	027300	000000		0			;0 WORDS
8369	027302	000000		0			;0 IS BUFF ADDRESS
8370	027304	000		.BYTE	0		;SECTOR 0
8371	027305	000		.BYTE	0		;TRACK 0
8372	027306	000000		0			;CYLINDER 0
8373							
8374	027310	104417		TLOADRK			;LOAD RK REGS
8375	027312	104423		TWAT16			;WAIT FOR INTERRUPT
8376	027314	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
8377							
8378	027316	004437	035530	JSR	R4,LRLOAD		;LOAD "L" REGS
8379	027322	000127		WRHEAD			;WRHEAD
8380	027324	177777		-1			; -1 WORDS
8381	027326	063414		OBUF			;OBUF IS BUFF ADDRESS

B13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 156
T104 FILE UNSAFE

SEQ 0156

8382	027330	000		.BYTE	0	;SECTOR 0
8383	027331	000		.BYTE	0	;TRACK 0
8384	027332	000000		0		;CYLINDER 0
8385	027334	012737	000040 001616	MOV	#DMD,L.MR1	;SET DIAGNOSTIC-MODE
8386						
8387	027342	104417		TLOADRK		;LOAD RK REGS
8388	027344	004437	036004	JSR	R4,MCLOCK	;CLOCK THROUGH SEEK AND DRIVE CLEAR
8389	027350	001064		1064		
8390						
8391	027352	052762	000200 000026	BIS	#MIND,RKMR1(R2)	;SET INDEX
8392						
8393	027360	004437	036004	JSR	R4,MCLOCK	;CLOCK INDEX
8394	027364	001001		1001		
8395						
8396	027366	042762	000200 000026	BIC	#MIND,RKMR1(R2)	;CLEAR INDEX
8397						
8398	027374	004437	036004	JSR	R4,MCLOCK	;CLOCK CLEAR
8399	027400	001001		1001		
8400						
8401	027402	005062	000026	CLR	RKMR1(R2)	;CLEAR DIAGNOSTIC MODE
8402						
8403	027406	104426		TWAIT64		;WAIT FOR INTERRUPT
8404	027410	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8405						
8406	027412	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8407	027414	104004		ERROR	4 ;OR 5, 6, 7, 10	;REPORT ALL ERRORS
8408						
8409	027416	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
8410	027422	000101		SELDRV		;SELDRV
8411	027424	000000		0		;0 WORDS
8412	027426	000000		0		;0 IS BUFF ADDRESS
8413	027430	000		.BYTE	0	;SECTOR 0
8414	027431	000		.BYTE	0	;TRACK 0
8415	027432	000000		0		;CYLINDER 0
8416						
8417	027434	005037	001616	CLR	L.MR1	;CLEAR DIAG MODE
8418						
8419	027440	104417		TLOADRK		;LOAD RK REGS
8420	027442	104423		TWAIT6		;WAIT FOR INTERRUPT
8421	027444	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8422						
8423	027446	104422		TCHKWE		;CHECK OPERATION WITH EXPECTED ERROR
8424	027450	040400		UNSEERR!DROTERR		;UNSAFE AND DRIVE OFF TRACK
8425	027452	000000		0		
8426	027454	000000		0		
8427	027456	104004		ERROR	4; OR 5,6,7	;REPORT ANY DISCREPANCIES
8428						
8429	027460	104416		TSSINIT		;CLEAR SUBSYSTEM
8430	027462	104003		ERROR	3	;BAD INIT ERROR
8431						
8432	027464	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
8433	027470	000101		SELDRV		;SELDRV
8434	027472	000000		0		;0 WORDS
8435	027474	000000		0		;0 IS BUFF ADDRESS
8436	027476	000		.BYTE	0	;SECTOR 0
8437	027477	000		.BYTE	0	;TRACK 0

8438	027500	000000		0		;CYLINDER 0
8439						
8440	027502	012737	000001 001616	MOV	#1,L.MR1	;SET MESSAGE SELECT ONE
8441						
8442	027510					
8443	027510	104417		TLOADRK		;LOAD RK REGS
8444	027512	104423		TWAIT6		;WAIT FOR INTERRUPT
8445	027514	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8446						
8447	027516	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8448	027520	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8449						
8450	027522	032737	000040 001574	BIT	#5.HDMM,T.MR2	;TEST IF HEADS HOME
8451	027530	001767		BEQ	15	
8452						
8453	027532	104416		TSSINIT		;CLEAR SUBSYSTEM
8454	027534	104003		ERROR	3	;BAD INIT ERROR
8455						
8456	027536	005037	001662	CLR	INTSET	;CLEAR INT FLAG
8457	027542	104434		TWAIT59		;WAIT FOR APPROX 160 MS
8458	027544	000240		NOP		;DON'T CARE ERROR RETURN
8459						
8460	027546	104416		TSSINIT		;CLEAR SUBSYSTEM
8461	027550	104003		ERROR	3	;BAD INIT ERROR
8462						
8463	027552	012762	000100 000000	MOV	#IE,RKCS1(R2)	;SET INTERRUPT ENABLE
8464						
8465	027560	104437		TWAIT8		;WAIT FOR SECOND INTERRUPT
8466	027562	104002		ERROR	2	
8467						
8468	027564	005037	001616	CLR	L.MR1	;CLEAR MR1
8469						
8470	027570	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
8471	027574	000105		CLEAR		;CLEAR
8472	027576	000000		0		;0 WORDS
8473	027600	000000		0		;0 IS BUFF ADDRESS
8474	027602	000		.BYTE	0	;SECTOR 0
8475	027603	000		.BYTE	0	;TRACK 0
8476	027604	000000		0		;CYLINDER 0
8477						
8478	027606	104417		TLOADRK		;LOAD RK REGS
8479	027610	104423		TWAIT6		;WAIT FOR INTERRUPT
8480	027612	104002		ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8481						
8482	027614	104421		TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8483	027616	104004		ERROR	4 ;OR 5, 6, 7	;REPORT ALL ERRORS
8484						
8485	027620	004437	041740	JSR	R4,GENCOM	;BUILD HEADERS
8486	027624	001200		1200		
8487						
8488	027626	004437	035530	JSR	R4,LRLoad	;LOAD "L" REGS
8489	027632	000127		WRHEAD		;WRHEAD
8490	027634	177676		-102		; -102 WORDS
8491	027636	063414		OBUFF		;OBUFF IS BUFF ADDRESS
8492	027640	000		.BYTE	0	;SECTOR 0
8493	027641	000		.BYTE	0	;TRACK 0

D13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 158
T104 FILE UNSAFE

SEQ 0158

```

8494 027642 000000      0      ;CYLINDER 0
8495
8496 027644 104417      TLOADRK      ;LOAD RK REGS
8497 027646 104426      TWTAT64      ;WAIT FOR INTERRUPT
8498 027650 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8499
8500 027652 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8501 027654 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8502
8503 *****
8504 *TEST 105 DUMMY TEST FOR PREVIOUS TEST EXIT
8505 * THIS TEST IS PRESENT TO MAKE $SWOBTB TABLE HAVE AN ENTRY
8506 * WHICH RELATES TO "NEWDRV". THIS IS NECESSARY IF AN ERROR OCCURS
8507 * IN THE PRECEDING TEST AND THAT ERROR ABORTS THE TEST.
8508 * IF THIS TEST WERE NOT PRESENT, THE PROGRAM WOULD SKIP THE
8509 * "NEWDRV" ROUTINE AND GO TO THE TEST FOLLOWING "NEWDRV".
8510 *
8511 * IN ADDITION, THE DRIVE IS CLEARED AND THE HEADS ARE ALLOWED
8512 * TO RELOAD. THIS MUST BE DONE TO PREVENT UNEXPECTED INTERRUPTS
8513 * FROM THE DRIVE COMING READY AT A LATER TIME.
8514 *****
8515 027656 000004      ST105: SCOPE
8516 027660 012737 000001 001262      MOV #1,STIMES      ;;DO 1 ITERATION
8517
8518 027666 104416      TSSINIT      ;CLEAR SUBSYSTEM
8519 027670 104003      ERROR 3      ;BAD INIT ERROR
8520
8521 027672 013762 001626 000010      MOV DRVNUM,RKCS2(R2) ;LOAD DRIVE NUMBER
8522 027700 012762 000001 000000      MOV #1,RKCS1(R2) ;SELECT THE DRIVE
8523 027706 032762 000200 000012 15: BIT #DRDY,RKDS(R2) ;TEST IF DRIVE READY
8524 027714 001774      BEQ 15      ;NO LOOP
8525
8526 027716 104416      TSSINIT      ;CLEAR SUBSYSTEM
8527 027720 104003      ERROR 3      ;BAD INIT ERROR
8528
8529 027722 004437 041740      JSR R4,GENCOM      ;GENERATE HEADERS FOR CYL 0
8530 027726 001200
8531
8532 027730 004437 035530      JSR R4,LRLOAD      ;LOAD "L" REGS
8533 027734 000127      WRHEAD      ;WRHEAD
8534 027736 177676      -102      ;-102 WORDS
8535 027740 063414      OBUFF      ;OBUFF IS BUFF ADDRESS
8536 027742 000      .BYTE 0      ;SECTOR 0
8537 027743 000      .BYTE 0      ;TRACK 0
8538 027744 000000      0      ;CYLINDER 0
8539
8540 027746 104417      TLOADRK      ;LOAD RK REGS
8541 027750 104426      TWTAT64      ;WAIT FOR INTERRUPT
8542 027752 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8543
8544 027754 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8545 027756 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8546 027760 005037 001676      CLR REFM      ;CLEAR REFORMAT SWITCH

```

```

8547      .SBTTL  **MULTI-DRIVE OPERATIONS
8548      NEWDRV: SCOPE
8549      MOV     #1,STIMES      ;DO ONLY ONCE
8550      BIT     #BIT7,DRVBIT    ;WERE WE TESTING DRIVE 7?
8551      BNE     3$             ;YES-SKIP
8552
8553      1$:     INC     DRVNUM    ;BUMP TO NEXT SEQUENTIAL ADDRESS
8554      ASL     DRVBIT          ;BUMP DRIVEBIT TO THAT POSITION
8555      BIT     DRVBIT,$DEVN     ;IS THIS DRIVE TO BE TESTED?
8556      BNE     2$             ;YES-EXIT
8557      BIT     #BIT8,DRVBIT    ;ALL DRIVES TESTED?
8558      BNE     3$             ;YES-EXIT
8559      BR      1$             ;ELSE CHECK NEXT DRIVE AVAILABLE
8560
8561      2$:     MOVB    #4,STSTNM ;SET TEST NUMBER FOR REPORTS
8562      JMP     TSTLUP          ;GO TO TEST LOOP TO CHECK THIS DRIVE
8563      3$:     CLR     DRVBIT    ;CLEAR DRIVE BIT
8564      CLR     DRVNUM          ;CLEAR DRIVE NUMBER
8565
8566      ;*****
8567      ;*TEST 106      RESET ATTENTIONS WITH UNIBUS INIT
8568      ;*
8569      ;*      DO A RECALIBRATE ON ALL AVAILIABLE DRIVES.
8570      ;*      ISSUE A RESET.  MAKE SURE ALL ATTENTION RESET.
8571      ;*
8572      ;*****
8573      TST106: SCOPE
8574      MOV     #10.,STIMES     ;DO 10. ITERATIONS
8575      CLR     R0              ;CLEAR DRIVE POSITION COUNTER
8576      MOV     #1,R1          ;PRESET BIT FOR POSITION 0 IN TESTING FOR AVAIL
8577      MOV     $DEVN,R3        ;GET DEVICE MAP
8578      TSSINIT          ;CLEAR SUBSYSTEM
8579      ERROR    3              ;BAD INIT ERROR
8580      1$:     BIT     R1,R3    ;TEST IF THIS DRIVE AVAILABLE
8581      BNE     2$             ;YES-SKIP TO SEEK
8582      3$:     ASL     R1        ;SHIFT DRIVE SELECT BIT
8583      INC     R0              ;BUMP DRIVE POSITION COUNTER
8584      BIT     #BIT8,R1        ;ALL DRIVE POSITIONS CHECKED
8585      BEQ     1$             ;NO-LOOP
8586      BR      4$             ;SKIP TO RESET
8587
8588      2$:     MOV     R0,L.CS2  ;LOAD DRIVE NUMBER
8589      MOV     #RECAL,L.CS1    ;LOAD RECALIBRATE
8590
8591      TLOADRK          ;LOAD RK REGS
8592      TWAT16          ;WAIT FOR INTERRUPT
8593      ERROR    2        ;TO SLOW/NOT COMPLETE ERROR
8594
8595      CLR     INTSET          ;CLEAR INTERRUPT FLAG
8596      MOV     #500.,R5        ;SET COUNT FOR 8 SECONDS
8597      MOV     #12,RKCS1(R2)   ;RESET INTERRUPT ENABLE
8598      MOV     RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8599      MOVB    T.ASOF+1,R4     ;ADJUST FOR CHECK OF ATTENTION
8600      BIC     #177400,R4      ;CLEAR UNUSED BITS
8601      BIT     R1,R4           ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8602      BNE     10$            ;YES - SKIP

```

F13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 160
T106 RESET ATTENTIONS WITH UNIBUS INIT

SEQ 0160

```

8603
8604 030206 104423          TWAT16          ;WAIT FOR 16 MS
8605 030210 000240          NOP              ;DON'T CARE RETURNS
8606 030212 000240          NOP
8607 030214 005305          DEC      R5        ;TATOL WAIT TIME IS 8 SECONDS
8608 030216 001362          BNE      12$      ;CHECK ATTENTION EACH 16 MS
8609 030220 104002          ERROR    2        ;REPORT IF NO ATTENTION IN 8 SEC
8610
8611 030222          10$:          TCHKOP          ;CHECK OPERATION FOR ANY ERRORS
8612 030222 104421          ERROR    4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8613 030224 104004
8614
8615 030226 000731          BR      3$          ;LOOP FOR NEXT DRIVE
8616 030230 000005          4$:      RESET        ;UNIBUS RESET
8617 030232 004737 044626          JSR      PC,$TKINT ;RESET KEYBOARD INTERRUPT ENABLE
8618
8619 030236 012701 000031          MOV      #25.,R1      ;DO A SHORT DELAY
8620 030242 005301          5$:      DEC      R1
8621 030244 001376          BNE      5$
8622 030246 004737 034434          JSR      PC,OPTTST ;SET UP OPTIONS
8623
8624 030252 104420          TGETRK          ;GET RK611 REGS
8625
8626 030254 105737 001557          TSTB      T.ASOF+1      ;ALL ATTENTION RESET?
8627 030260 001407          BEQ      6$          ;YES-SKIP
8628
8629 030262 012737 056526 001470          MOV      #EMDA,EM12N ;"DRIVE ATT NOT RESET RESULT OF
8630 030270 012737 056652 061010          MOV      #EMUR,DF011A ;UNIBUS RESET"
8631 030276 104012          ERROR    12
8632
8633 030300          6$:
8634          ;*****
8635          ;TEST 107      RESET ATTENTIONS WITH SUBSYSTEM CLEAR
8636          ;
8637          ;DO A RECALIBRATE ON ALL AVAILABLE DRIVES.
8638          ;ISSUE A SUBSYSTEM CLEAR. MAKE SURE ALL ATTENTIONS
8639          ;RESET.
8640          ;
8641          ;*****
8642 030300 000004          7ST107: SCOPE
8643 030302 012737 000012 001262          MOV      #10.,$TIMES ;DO 10. ITERATIONS
8644 030310 005000          CLR      R0          ;CLEAR DRIVE POSITION COUNTER
8645 030312 012701 000001          MOV      #1,R1      ;PRESET TO TEST POSITION 0
8646 030316 013703 001354          MOV      $CEVM,R3      ;CUT DEVICE MAP
8647 030322 104416          TSSINIT          ;CLEAR SUBSYSTEM
8648 030324 104003          ERROR    3          ;BAD INIT ERROR
8649 030326 030103          1$:      BIT      R1,R3      ;THIS DRIVE AVAILABLE?
8650 030330 001006          BNE      2$          ;YES-SKIP TO SEEK
8651 030332 006301          3$:      ASL      R1          ;SHIFT TO NEXT DRIVE POSITION
8652 030334 005200          INC      R0          ;DUMP POSITION COUNTER
8653 030336 032701 000400          BIT      #BIT8,R1      ;ALL POSITIONS CHECKED
8654 030342 001771          BEQ      1$          ;NO-LOOP
8655 030344 000441          BR      4$          ;YES-SKIP TO CLEAR
8656
8657 030346 010037 001610          2$:      MOV      R0,L.CS2      ;LOAD DRIVE NUMBER
8658 030352 012737 000113 001600          MOV      #RECAL,L.CS1 ;LOAD RECALIBRATE

```

G13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 161
T107 RESET ATTENTIONS WITH SUBSYSTEM CLEAR

SEQ 0161

```

8659 030360 104417      TLOADRK      ;LOAD RK REGS
8660 030362 104423      TWAT16      ;WAIT FOR INTERRUPT
8661 030364 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8662
8663 030366 005037 001662      CLR      INTSET      ;CLEAR INT FLAG
8664 030372 012705 000764      MOV      #500,R5      ;SET COUNT FOR 8 SECONDS
8665 030376 012762 000012 000000      MOV      #12,RKCS1(R2) ;RESET INTERRUPT ENABLE
8666 030404 016237 000016 001556 12$:  MOV      RKASOF(R2),T.ASOF ;GET ATTENTION REGISTER
8667 030412 113704 001557      MOV      T.ASOF+1,R4 ;ADJUST FOR CHECK OF ATTENTION
8668 030416 042704 177400      BIC      #177400,R4 ;CLEAR UNUSED BITS
8669 030422 030104      BIT      R1,R4 ;CHECK IF ATT SET FROM DRIVE RECAL'ED
8670 030424 001006      BNE      10$ ;YES - SKIP
8671
8672 030426 104423      TWAT16      ;WAIT FOR 16 MS
8673 030430 000240      NOP ;DON'T CARE RETURNS
8674 030432 000240      NOP
8675 030434 005305      DEC      R5 ;TOTAL WAIT TIME IS 8 SECONDS
8676 030436 001362      BNE      12$ ;CHECK ATTENTION EACH 16 MS
8677 030440 104002      ERROR 2 ;REPORT IF NO ATTENTION IN 8 SEC
8678
8679 030442 10$:          TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8680 030442 104421      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8681 030444 104004
8682
8683 030446 000731      BR      3$ ;LOOP FOR NEXT DRIVE
8684
8685 030450 052762 000040 000010 4$:  BIS      #SCLR,RKCS2(R2) ;DO SUBSYSTEM CLEAR
8686 030456 012701 000031      MOV      #25,R1 ;DO A SHORT DELAY
8687 030462 005301 5$:      DEC      R1
8688 030464 001376      BNE      5$
8689
8690 030466 104420      TGETRK      ;GET RK611 REGS
8691
8692 030470 105737 001557      TSTB      T.ASOF+1 ;TEST ALL ATTENTION RESET
8693 030474 001407      BEQ      6$ ;YES-SKIP
8694
8695 030476 012737 056526 001470      MOV      #EMDA,EM12N ;"DRIVE ATT NOT RESET AS RESULT OF
8696 030504 012737 056742 061010      MOV      #EMSCLR,DF011A ;SUBSYSTEM CLEAR"
8697 030512 104012      ERROR 12
8698
8699 030514 6$:          ;*****
8700 ;*****
8701 ;TEST 110 SVAL AND ATTENTION FROM OTHER DRIVE
8702 ;
8703 ; DO A RECALIBRATE ON ONE AVAILABLE DRIVE. DO A SELECT
8704 ; ON ANOTHER AVAILABLE DRIVE. MAKE SURE STATUS VALID
8705 ; IS SET. WAIT FOR SECOND INTERRUPT FROM RECALIBRATE
8706 ; MAKE SURE STATUS VALID REMAINS SET AND DRIVE STATUS
8707 ; CHANGE REMAINS RESET.
8708 ;
8709 ; REPEAT FOR ALL COMBINATIONS OF TWO AVAILABLE DRIVES.
8710 ;
8711 ; NOTE: THIS TEST WILL ONLY BE DONE IF AT LEAST
8712 ; TWO DRIVES ARE AVAILABLE.
8713 ;*****
8714 ;*****

```

H13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 162
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0162

8715	030514	000004			TST110: SCOPE	
8716	030516	012737	000012	001262	MOV #10, STIMES	; DO 10. ITERATIONS
8717	030524	013746	001354		MOV \$DEVN, -(SP)	; PUT DEVICE MAP ON STACK
8718	030530	004437	035730		JSR R4, BITCNT	; COUNT NUMBER OF BITS (# OF DRIVES)
8719	030534	022627	000001		CMP (SP)+, #1	; COMPARE TO 1
8720	030540	101007			BHI 2\$	; SKIP IF MORE THAN 1
8721	030542	005737	001304		TST \$PASS	; CHECK IF PASS 0
8722	030546	001002			BNE 1\$	; NO-SKIP
8723						
8724	030550	104401	052562		TYPE OPR018	; "OVERLAPPED OPERATION BYPASSED"
8725	030554	000137	031770		1\$: JMP \$EOP	; GET OUT.
8726						
8727	030560	012737	177777	001224	2\$: MOV #-1, STMP1	; SET LOOP CONTROL FLAG
8728	030566	013705	001354		MOV \$DEVN, R5	; GET DEVICE MAP
8729	030572	005000			CLR R0	; CLEAR FOR DRIVE #A
8730	030574	005001			CLR R1	; CLEAR FOR DRIVE #B
8731	030576	012703	000001		MOV #1, R3	; SET DRIVE POSITION A
8732	030602	012704	000001		MOV #1, R4	; SET DRIVE POSITION B
8733	030606	012737	030616	001110	MOV #3\$, \$LPERR	; SET LOCAL LOOP ON ERROR
8734	030614	000477			BR 11\$	; GO SET UP POINTERS
8735						
8736	030616				3\$: TSSINIT	; CLEAR SUBSYSTEM
8737	030616	104416			ERROR 3	; BAD INIT ERROR
8738	030620	104003				
8739						
8740	030622	010037	001610		MOV R0, L.CS2	; LOAD DRIVE A
8741	030626	010037	001626		MOV R0, DRVNUM	; LOAD FOR REPORT
8742	030632	012737	000113	001600	MOV #RECAL, L.CS1	; LOAD RECAL
8743						
8744	030640	104417			TLOADRK	; LOAD RK REGS
8745	030642	104423			TWAIT6	; WAIT FOR INTERRUPT
8746	030644	104002			ERROR 2	; TO SLOW/NOT COMPLETE ERROR
8747	030646	104421			TCHKOP	; CHECK OPERATION FOR ANY ERRORS
8748	030650	104004			ERROR 4 ;OR 5, 6, 7	; REPORT ALL ERRORS
8749						
8750	030652	005037	001662		CLR INTSET	; CLEAR INT FLAG
8751						
8752	030656	010137	001610		MOV R1, L.CS2	; LOAD DRIVE B
8753	030662	010137	001626		MOV R1, DRVNUM	; LOAD FOR REPORT
8754	030666	012737	000101	001600	MOV #SELDRV, L.CS1	; LOAD DRIVE SELECT
8755						
8756	030674	104417			TLOADRK	; LOAD RK REGS
8757	030676	104423			TWAIT6	; WAIT FOR INTERRUPT
8758	030700	104002			ERROR 2	; TO SLOW/NOT COMPLETE ERROR
8759						
8760	030702	104421			TCHKOP	; CHECK OPERATION FOR ANY ERRORS
8761	030704	104004			ERROR 4 ;OR 5, 6, 7	; REPORT ALL ERRORS
8762						
8763	030706	032737	100000	001552	BIT #SVAL, T.DS	; CHECK IF STATUS VALID SET
8764	030714	001007			BNE 4\$	; YES - SKIP
8765	030716	012737	056342	001460	MOV #EMSVAL, EM11N	; "STATUS VALID NOT SET RESULT OF
8766	030724	012737	047640	061010	MOV #OPEROO, DFO11A	; DRIVE SELECT"
8767	030732	104011			ERROR 11	
8768						
8769	030734	005037	001662		4\$: CLR INTSET	; CLEAR INT FLAG
8770	030740	104436			TWAIT2S	; WAIT FOR SEEK COMPLETE INTERRUPT

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 163
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0163

8771	030742	000401			BR	44\$		;NONE RECEIVED - SKIP
8772	030744	000406			BR	55\$		; RECEIVED - SKIP
8773								
8774	030746	010037	001626		44\$: MOV	RO,DRVNUM		;SET DRIVE FOR REPORT
8775	030752	012737	060377	001372	MOV	#DH017,DH2N		; "COMMAND - SELECT AFTER RECAL"
8776	030760	104702			ERROR	2		
8777								
8778	030762	104420			55\$: TGETRK			;GET RK REGS
8779	030764	032737	100000	001552	BIT	#SVAL,T.DS		;TEST IF SVAL STILL SET
8780	030772	001007			BNE	55		;YES - SKIP
8781								
8782	030774	012737	056342	001510	MOV	#EMSVAL,EM14N		; "STATUS VALID RESET RESULT OF
8783	031002	012737	060422	061010	MOV	#DH018,DF011A		;RECAL COMPLETE ATTENTION AFTER SEL"
8784	031010	104014			ERROR	14		
8785								
8786	031012	104415			55\$: SCOP1			;LOCAL LOOP TO 3\$
8787								
8788								
8789								
8790								
8791								
8792								
8793								
8794								
8795								
8796								
8797								
8798								
8799								
8800								
8801								
8802								
8803	031014	005237	001224		11\$: INC	\$TMP1		;INCREMENT PASS CONTROL
8804	031020	001024			BNE	16\$		;SKIP IF NOT ZERO
8805								; (IT WILL BE ZERO ON THE 1ST PASS)
8806								
8807	031022	030305			12\$: BIT	R3,R5		;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8808	031024	001006			BNE	13\$		;YES-SKIP
8809								
8810	031026	005200			22\$: INC	RO		;BUMP RO (DRIVE A)
8811	031030	006303			ASL	R3		;SHIFT DRIVE SELECT BIT ONE POSITION
8812	031032	032703	000400		BIT	#BIT8,R3		;IF BIT 8 IS SET, ALL DRIVES HAVE
8813	031036	001771			BEQ	12\$		;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8814	031040	000464			BR	50\$		;DONE-EXIT
8815								
8816	031042	010001			13\$: MOV	RO,R1		;SET DRIVE B TO THE SAME AS A
8817	031044	010304			MOV	R3,R4		
8818	031046	005201			14\$: INC	R1		;BUMP R1 (DRIVE B)
8819	031050	006304			ASL	R4		;SHIFT SELECTOR BIT ONE POSITION
8820	031052	030405			BIT	R4,R5		;IS THIS DRIVE AVAIL?
8821	031054	001004			BNE	15\$		;YES-SKIP
8822	031056	032704	000400		BIT	#BIT8,R4		;WERE ALL POSITIONS CHECKED?
8823	031062	001771			BEQ	14\$		;NO-LOOP
8824	031064	000452			BR	50\$		;DONE-EXIT
8825								
8826	031066	000137	030616		15\$: JMP	3\$		;GO DO THE TEST ON THE DRIVE A & B

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

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

THIS TECHNIQUE IS CONTINUED UNTIL ALL COMBINATIONS ARE CHECKED.

J13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 164
T110 SVAL AND ATTENTION FROM OTHER DRIVE

SEQ 0164

```

8827
8828 031072 032737 000001 001224 16$: BIT #BIT0,$TMP1 ;CONTAINED IN R0 & R1
8829 031100 001410 BEQ 17$ ;IS PASS FLAGS ODD?
8830 ;NO-SKIP
8831 031102 010046 MOV R0,-(SP) ;
8832 031104 010346 MOV R3,-(SP) ;SWAP R0 & R1, R3 & R4
8833 031106 010403 MOV R4,R3 ;TO EXCHANGE DRIVE A & B
8834 031110 010100 MOV R1,R0 ;
8835 031112 012604 MOV (SP)+,R4 ;
8836 031114 012601 MOV (SP)+,R1 ;
8837 031116 000137 030616 JMP 3$ ;REPEAT TEST ON THIS COMBO.
8838
8839 031122 032737 000002 001224 17$: BIT #BIT1,$TMP1 ;TEST IF PASS FLAGS AT HALF MODULE 4?
8840 031130 001410 BEQ 19$ ;NO-SKIP TO BUMP DRIVE B
8841 031132 005200 18$: INC R0 ;BUMP DRIVE A
8842 031134 006303 ASL R3 ;SHIFT DRIVE SELECT BIT
8843 031136 030305 BIT R3,R5 ;AVAILABLE?
8844 031140 001014 BNE 20$ ;YES-SKIP
8845 031142 032703 000400 BIT #BIT8,R3 ;ALL CHECKED?
8846 031146 001771 BEQ 18$ ;NO-SKIP
8847 031150 000412 BR 21$ ;GO TO NEXT PASS
8848
8849 031152 005201 19$: INC R1 ;BUMP DRIVE B
8850 031154 006304 ASL R4 ;SHIFT DRIVE SELECT BIT
8851 031156 030405 BIT R4,R5 ;AVAILABLE?
8852 031160 001004 BNE 20$ ;YES-SKIP
8853 031162 032704 000400 BIT #BIT8,R4 ;ALL CHECKED?
8854 031166 001771 BEQ 19$ ;NO-LOOP
8855 031170 000404 BR 23$ ;YES-SKIP TO NEXT PASS
8856
8857 031172 000137 030616 20$: JMP 3$ ;GO TEST THIS COMBO
8858
8859 031176 010100 21$: MOV R1,R0 ;SET DRIVE 0 TO LOW POSITION THIS PASS
8860 031200 010403 MOV R4,R3 ;SET SELECT BITS TO AGREE
8861 031202 005037 001224 23$: CLR $TMP1 ;CLEAR PASS FLAGS
8862 031206 000137 031026 JMP 22$ ;GO SET UP A & B
8863 031212 50$:
8864 ;*****
8865 ;TEST 111 OVERLAPPED OPERATIONS
8866 ;
8867 ; DO A RECALIBRATE ON BOTH DRIVE A AND DRIVE B. ISSUE A
8868 ; SEEK ON DRIVE A TO CYLINDER 1. IMMEDIATELY ISSUE A WRITE
8869 ; DATA TO CYLINDER 100, TRACK 0, HEAD 0 ON DRIVE B.
8870 ; AT THE END OF THE DATA TRANSFER NO ERRORS SHOULD
8871 ; BE SET AND DRIVE A HAS ATTENTION SET.
8872 ;
8873 ; REPEAT FOR ALL COMBINATIONS OF TWO DRIVES.
8874 ;
8875 ; NOTE: IF ONLY ONE DRIVE IS AVAILABLE THE
8876 ; TEST WILL NOT BE DONE.
8877 ;
8878 ;*****
8879 031212 000004 1ST111: SCOPE
8880 031214 012737 000005 001262 MOV #5,$TIMES ;;DO 5. ITERATIONS
8881
8882 031222 012737 177777 001224 2$: MOV #-1,$TMP1 ;SET LOOP CONTROL FLAG

```

K13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 165
T111 OVERLAPPED OPERATIONS

SEQ 0165

8883	031230	013705	001354		MOV	\$DEVN,R5	;GET DEVICE MAP
8884	031234	005000			CLR	R0	;CLEAR FOR DRIVE #A
8885	031236	005001			CLR	R1	;CLEAR FOR DRIVE #B
8886	031240	012703	000001		MOV	#1,R3	;SET DRIVE POSITION A
8887	031244	012704	000001		MOV	#1,R4	;SET DRIVE POSITION B
8888	031250	012737	031260	001110	MOV	#3\$,SLPERR	;SET LOCAL LOOP ON ERROR
8889	031256	000545			BR	11\$	;GO SET UP POINTERS
8890	031260						
8891	031260	104416			TSSINIT		;CLEAR SUBSYSTEM
8892	031262	104003			ERROR	3	;BAD INIT ERROR
8893							
8894	031264	010037	001626		MOV	R0,DRVNUM	;STORE DRIVE FOR REPORT
8895	031270	010037	001610		MOV	R0,L.CS2	;SETUP DRIVE A TO RECAL
8896	031274	012737	000113	001600	MOV	#RECAL,L.CS1	
8897							
8898	031302	104417			TLOADRK		;LOAD RK REGS
8899	031304	104423			TWAT16		;WAIT FOR INTERRUPT
8900	031306	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8901	031310	005037	001662		CLR	INTSET	;CLEAR INTERRUPT FLAG
8902							
8903	031314	104437			TWAT8S		;WAIT FOR SECOND INTERRUPT
8904	031316	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8905							
8906	031320	012737	000105	001600	MOV	#CLEAR,L.CS1	;SET UP TO CLEAR DRIVE
8907	031326	104417			TLOADRK		;LOAD RK REGS
8908	031330	104423			TWAT16		;WAIT FOR INTERRUPT
8909	031332	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8910	031334						
8911	031334	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8912	031336	104004			ERROR	4 ;OR 5 6, 7	;REPORT ALL ERRORS
8913	031340	032737	040000	001540	BIT	#DI,T.CS1	;TEST IF DI STILL SET
8914	031346	001372			BNE	4\$	;YES - LOOP
8915	031350	010137	001626		MOV	R1,DRVNUM	;STORE DRIVE FOR REPORT
8916	031354	010137	001610		MOV	R1,L.CS2	;SETUP DRIVE B TO RECAL
8917	031360	012737	000113	001600	MOV	#RECAL,L.CS1	
8918							
8919	031366	104417			TLOADRK		;LOAD RK REGS
8920	031370	104423			TWAT16		;WAIT FOR INTERRUPT
8921	031372	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8922	031374	005037	001662		CLR	INTSET	;CLEAR INTERRUPT FLAG
8923	031400	104437			TWAT8S		;WAIT FOR SECOND INTERRUPT
8924	031402	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8925	031404	012737	000105	001600	MOV	#CLEAR,L.CS1	;SET UP DRIVE CLEAR
8926	031412	104417			TLOADRK		;LOAD RK REGS
8927	031414	104423			TWAT16		;WAIT FOR INTERRUPT
8928	031416	104002			ERROR	2	;TO SLOW/NOT COMPLETE ERROR
8929	031420						
8930	031420	104421			TCHKOP		;CHECK OPERATION FOR ANY ERRORS
8931	031422	104004			ERROR	4 ;OR 5 6, 7	;REPORT ALL ERRORS
8932	031424	032737	040000	001540	BIT	#DI,T.CS1	;TEST IF DI STILL SET
8933	031432	001372			BNE	5\$	;YES - LOOP
8934							
8935	031434	010037	001626		MOV	R0,DRVNUM	;STORE DRIVE FOR REPORT
8936	031440	010037	001610		MOV	R0,L.CS2	;SETUP DRIVE A TO SEEK
8937	031444	012737	000001	001614	MOV	#1,L.DCYL	;TO CYL 1
8938	031452	012737	000117	001600	MOV	#SEEK,L.CS1	

L13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 166
T111 OVERLAPPED OPERATIONS

SEQ 0166

```

8939
8940 031460 104417      TLOADRK      ;LOAD RK REGS
8941 031462 104423      TWAT16      ;WAIT FOR INTERRUPT
8942 031464 104002      ERROR 2      ;TC SLOW/NOT COMPLETE ERROR
8943
8944 031466 010137 001626  MOV R1,DRVNUM ;STORE DRIVE FOR REPORT
8945 031472 010137 001610  MOV R1,L.CS2 ;SETUP DRIVE B TO WRITE DATA
8946 031476 012737 000100 001614  MOV #100,L.DCYL ;AT CYL 100
8947 031504 012737 177400 001602  MOV #-400,L.WC ;400 WORDS
8948 031512 012737 063414 001604  MOV #0BUFF,L.BA
8949 031520 012737 000123 001600  MOV #WRDATA,L.CS1
8950
8951 031526 104417      TLOADRK      ;LOAD RK REGS-DO WRITE
8952
8953 031530 104427      TWAT80      ;WAIT FOR INTERRUPT
8954 031532 104002      ERROR 2      ;TO SLOW/NOT COMPLETE ERROR
8955
8956 031534 104421      TCHKOP      ;CHECK OPERATION FOR ANY ERRORS
8957 031536 104004      ERROR 4 ;OR 5, 6, 7 ;REPORT ALL ERRORS
8958
8959 031540 010037 001626  MOV R0,DRVNUM ;STORE DRIVE FOR REPORT
8960 031544 130337 001557  BITB R3,T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS DRIVE A
8961 031550 001007      BNE 10$      ;YES-SKIP
8962 031552 012737 056526 001460  MOV #EMDA,EMI1N ;"DRIVE ATTENTION NOT SET RESULT OF
8963 031560 012737 047752 061010  MOV #EMSK,DF011A ;SEEK"
8964 031566 104011      ERROR 11
8965
8966 031570 104415      10$: SCOP1      ;LOCAL LOOP TO 3$
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983 031572 005237 001224      11$: INC STMP1      ;INCREMENT PASS CONTROL
8984 031576 001024      BNE 16$      ;SKIP IF NOT ZERO
8985
8986
8987
8988
8989
8990 031600 030305      12$: BIT R3,R5      ;TEST IF BIT POSITION FOR A AT AVAIL DRIVE
8991 031602 001006      BNE 13$      ;YES-SKIP
8992
8993
8994 031604 005200      22$: INC R0      ;BUMP R0 (DRIVE A)
8995 031606 006303      ASL R3      ;SHIFT DRIVE SELECT BIT ONE POSITION
8996 031610 032703 000400  BIT #BIT8,R3 ;IF BIT 8 IS SET, ALL DRIVES HAVE
8997 031614 001771      BEQ 12$      ;BEEN CHECKED; IF NOT CHECK NEXT POSITION
8998 031616 000464      BR 50$      ;DONE-EXIT

```

M13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 167
T111 OVERLAPPED OPERATIONS

SEQ 0167

8995											
8996	031620	010001			13\$:	MOV	R0,R1				;SET DRIVE B TO THE SAME AS A
8997	031622	010304				MOV	R3,R4				
8998	031624	005201			14\$:	INC	R1				;BUMP R1 (DRIVE B)
8999	031626	006304				ASL	R4				;SHIFT SELECTOR BIT ONE POSITION
9000	031630	030405				BIT	R4,R5				;IS THIS DRIVE AVAIL?
9001	031632	001004				BNE	15\$				;YES-SKIP
9002	031634	032704	000400			BIT	#BIT8,R4				;WERE ALL POSITIONS CHECKED?
9003	031640	001771				BEQ	14\$				;NO-LOOP
9004	031642	000452				BR	50\$				;DONE-EXIT
9005											
9006	031644	000137	031260		15\$:	JMP	3\$				;GO DO THE TEST ON THE DRIVE A & B
9007											;CONTAINED IN R0 & R1
9008	031650	032737	000001	001224	16\$:	BIT	#BIT0,\$TMP1				;IS PASS FLAGS ODD?
9009	031656	001410				BEQ	17\$				;NO-SKIP
9010											
9011	031660	010046				MOV	R0,-(SP)				
9012	031662	010346				MOV	R3,-(SP)				;SWAP R0 & R1, R3 & R4
9013	031664	010403				MOV	R4,R3				;TO EXCHANGE DRIVE A & B
9014	031666	010100				MOV	R1,R0				
9015	031670	012604				MOV	(SP)+,R4				
9016	031672	012601				MOV	(SP)+,R1				
9017	031674	000137	031260			JMP	3\$				;REPEAT TEST ON THIS COMBO.
9018											
9019	031700	032737	000002	001224	17\$:	BIT	#BIT1,\$TMP1				;TEST IF PASS FLAGS AT HALF MODULE 4?
9020	031706	001410				BEQ	19\$				;NO-SKIP TO BUMP DRIVE B
9021	031710	005200			18\$:	INC	R0				;BUMP DRIVE A
9022	031712	006303				ASL	R3				;SHIFT DRIVE SELECT BIT
9023	031714	030305				BIT	R3,R5				;AVAILABLE?
9024	031716	001014				BNE	20\$				;YES-SKIP
9025	031720	032703	000400			BIT	#BIT8,R3				;ALL CHECKED?
9026	031724	001771				BEQ	18\$				;NO-SKIP
9027	031726	000412				BR	21\$				;GO TO NEXT PASS
9028											
9029	031730	005201			19\$:	INC	R1				;BUMP DRIVE B
9030	031732	006304				ASL	R4				;SHIFT DRIVE SELECT BIT
9031	031734	030405				BIT	R4,R5				;AVAILABLE?
9032	031736	001004				BNE	20\$				;YES-SKIP
9033	031740	032704	000400			BIT	#BIT8,R4				;ALL CHECKED?
9034	031744	001771				BEQ	19\$				;NO-LOOP
9035	031746	000404				BR	23\$				;YES-SKIP TO NEXT PASS
9036											
9037	031750	000137	031260		20\$:	JMP	3\$				;GO TEST THIS COMBO
9038											
9039	031754	010100			21\$:	MOV	R1,R0				;SET DRIVE 0 TO LOW POSITION THIS PASS
9040	031756	010403				MOV	R4,R3				;SET SELECT BITS TO AGREE
9041	031760	005037	001224		23\$:	CLR	\$TMP1				;CLEAR PASS FLAGS
9042	031764	000137	031604			JMP	22\$				;GO SET UP A & B
9043	031770				50\$:						

N13

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 168
END OF PASS ROUTINE

SEQ 0168

```

9044 .SBTTL END OF PASS ROUTINE
9045
9046 ;*****
9047 ;*INCREMENT THE PASS NUMBER ($PASS)
9048 ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
9049 ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
9050 ;*IF THERES A MONITOR GO TO IT
9051 ;*IF THERE ISN'T JUMP TO TST1
9052
9053 $EOP:
9054 SCOPE
9055 CLR $TSTNM ;ZERO THE TEST NUMBER
9056 CLR $TIMES ;ZERO THE NUMBER OF ITERATIONS
9057 INC $PASS ;INCREMENT THE PASS NUMBER
9058 BIC #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
9059 DEC (PC)+ ;LOOP?
9060 $EOPCT: .WORD 1
9061 BGT $DOAGN ;YES
9062 MOV (PC)+,(PC)+ ;RESTORE COUNTER
9063 $ENDCT: .WORD 1
9064 TYPE $EOPCT
9065 BR 65$ ;TYPE ASCIZ STRING
9066 ;65$: .ASCIZ <12><15>/END PASS #/ ;GET OVER THE ASCIZ
9067 64$:
9068 MOV $PASS,-(SP) ;SAVE $PASS FOR TYPEOUT
9069 ;TYPE PASS NUMBER
9070 TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
9071 TYPE 67$ ;TYPE ASCIZ STRING
9072 BR 66$ ;GET OVER THE ASCIZ
9073 ;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
9074 66$:
9075 MOV $ERTTL,-(SP) ;SAVE $ERTTL FOR TYPEOUT
9076 ;TOTAL NUMBER OF ERRORS
9077 TYPDS ;GO TYPE--DECIMAL ASCII WITH SIGN
9078 TYPE $CRLF ;TYPE CARRIAGE RETURN, LINE FEED
9079 CLR $ERTTL ;CLEAR ERROR TOTAL
9080 $GET42: MOV #42,R0 ;GET MONITOR ADDRESS
9081 BEQ $DOAGN ;BRANCH IF NO MONITOR
9082 RESET ;CLEAR THE WORLD
9083 JSR PC,(R0) ;GO TO MONITOR
9084 NOP ;SAVE ROOM
9085 NOP ;FOR
9086 NOP ;ACT11
9087 $DOAGN:
9088 JMP #42,(PC)+ ;RETURN
9089 $RTNAD: .WORD TST1
9090 $ENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING
9091 .EVEN
9092 .SBTTL ROUTINE TO SIZE MEMORY
9093
9094 ;*****
9095 ;CALL:
9096 ; JSR PC,$SIZE
9097 ; RETURN
9098 ;$LSTAD WILL CONTAIN:
9099

```

```

9100      ;*      WITH KT11 OPTION      -- LAST VIRTUAL ADDRESS OF THE LAST BANK
9101      ;*      WITHOUT KT11 OPTION  -- LAST ABSOLUTE ADDRESS OF AVAILABLE MEMORY
9102      ;*SLSTBK WILL CONTAIN THE LAST BANK AS A SAF
9103      ;*SKT11 IS THE MEMORY MANAGEMENT KEY
9104      ;*BIT07 = 0 DON'T USE MEMORY MANAGEMENT
9105      ;*      MUST BE SETUP BEFORE THE CALL
9106      ;*BIT15 = 0 DON'T HAVE MEMORY MANAGEMENT OPTION
9107      ;*      DETERMINED BY ROUTINE
9108
9109      $SIZE:  MOV     R0,-(SP)      ;:SAVE R0 ON THE STACK
9110             MOV     R1,-(SP)      ;:SAVE R1 ON THE STACK
9111             MOV     R2,-(SP)      ;:SAVE R2 ON THE STACK
9112             MOV     R3,-(SP)      ;:SAVE R3 ON THE STACK
9113             MOV     @#ERRVEC,-(SP) ;:SAVE PRESENT ERROR VECTOR PS & PC
9114             MOV     @#ERRVEC+2,-(SP)
9115             MOV     SP,R0          ;:SAVE THE STACK POINTER
9116      ;;SET THE ERRVEC PS TO THE PRESENT PS
9117      TRAP
9118             MOV     (SP)+,@#ERRVEC+2 ;:PUSH OLD PSW AND PC ON STACK
9119             MOV     @3776,R1        ;:SAVE THE PSW IN @#ERRVEC+2
9120             TSTB    (PC)+          ;:SETUP ADDRESS
9121             .WORD    200            ;:USE MEMORY MANAGEMENT?
9122             BPL     SCORE          ;:SET TO USE MEMORY MANAGEMENT
9123             MOV     @SKTNEX,@#ERRVEC ;:BR IF NO
9124             TST     @#SR0           ;:SET FOR TIMEOUT
9125             BIS     @100000,SKT11  ;:KT11 ARE YOU THERE?
9126             CLR     -(SP)          ;:YES--SET KT11 KEY
9127             MOV     @KIPAR0,R2      ;:INITIALIZE FOR "PAR" LOADING
9128             MOV     @108,R3         ;:ADDRESS OF FIRST "PAR"
9129             MOV     @77406,-40(R2)  ;:LOAD EIGHT "PAR.'S" AND EIGHT "PDR.'S"
9130             MOV     (SP),(R2)+      ;:PDR = 4K UP, READ/WRITE
9131             ADD     @200,(SP)       ;:LOAD "PAR"
9132             SOB     R3,1$           ;:UPDATE FOR NEXT "PAR"
9133             MOV     @177600,-(R2)   ;:LOOP UNTIL ALL EIGHT ARE LOADED
9134             CLR     -(R2)           ;:SETUP KIPAR7 FOR I/O
9135             MOV     @2$,@#ERRVEC    ;:SETUP KIPAR6 FOR TESTING
9136             MOV     @20,@#SR3      ;:CATCH TIMEOUT IF NO SR3
9137             BR      3$              ;:ENABLE 22 BIT MODE
9138             CMP     (SP)+,(SP)+     ;:THIS PDP-11 HAS A SR3 REGISTER
9139             INC     @#SR0           ;:CLEAN OFF THE STACK--NO SR3
9140             MOV     @SKTOUT,@#ERRVEC ;:TURN ON MEMORY MANAGEMENT
9141             TST     @#143776        ;:SET FOR TIME OUT
9142             ADD     @40,(R2)        ;:TRAP ON NON-EX-MEM
9143             CMP     @#KIPAR7,(R2)   ;:MAKE A 1K STEP
9144             BHI     4$              ;:LAST ONE?
9145             SKTOUT: MOV     (R2),R2 ;:NO--TRY IT
9146             CLR     @#SR0          ;:GET LAST BANK+1
9147             BR      $SIZE          ;:TURN OFF MEMORY MANAGEMENT
9148             BIC     @100000,SKT11  ;:KT11 NON-EXISTENT
9149             MOV     @SCROUT,@#ERRVEC ;:SET FOR TIMEOUT
9150             CLR     R2              ;:SET UP BANK
9151             ADD     @4000,R1         ;:INCREMENT BY 1K
9152             ADD     @40,R2          ;:1K STEP
9153             TST     (R1)            ;:TRAP ON TIME OUT
9154             CMP     @177776,R1      ;:LAST ONE
9155             BNE     1$              ;:NO--TRY AGAIN

```

9156	032436	162701	004000		SCRUT: SUB	#4000,R1	
9157	032442	162702	000040		\$SIZE: SUB	#40,R2	:: DROP BACK
9158	032446	010006			MOV	RO,SP	:: RESTORE THE STACK
9159	032450	012637	000006		MOV	(SP)+,2*ERRVEC+2	:: RESTORE ERROR VECTOR
9160	032454	012637	000004		MOV	(SP)+,2*ERRVEC	
9161	032460	010137	032502		MOV	R1,\$LASTAD	:: LAST ADDRESS
9162	032464	010237	032504		MOV	R2,\$LASTBK	:: LAST BANK
9163	032470	012603			MOV	(SP)+,R3	:: RESTORE R3
9164	032472	012602			MOV	(SP)+,R2	:: RESTORE R2
9165	032474	012601			MOV	(SP)+,R1	:: RESTORE R1
9166	032476	012600			MOV	(SP)+,RO	:: RESTORE RO
9167	032500	000207			RTS	PC	
9168	032502	000000			\$LASTAD: .WORD	0	:: CONTAINS THE LAST ADDRESS
9169	032504	000000			\$LASTBK: .WORD	0	:: CONTAINS THE LAST BANK
9170					.SBTTL	SCOPE HANDLER ROUTINE	
9171							
9172							
9173							
9174							
9175							
9176							
9177							
9178							
9179							
9180							
9181							
9182							
9183							
9184	032506						
9185	032506	104407			\$SCOPE:		
9186							
9187							
9188	032510	021627	001002				
9189	032514	101002					
9190	032516	000137	033526				
9191	032522	032777	040000	146410	1\$:	BIT	#BIT14,\$SWR
9192	032530	001131				BNE	\$OVER
9193							
9194	032532	000416					
9195							
9196	032534	013746	000004				
9197	032540	012737	032560	000004			
9198	032546	005737	177060				
9199	032552	012637	000004				
9200	032556	000500					
9201	032560	022626					
9202	032562	012637	000004		5\$:	CMP	(SP)+,(SP)+
9203	032566	000440				MOV	(SP)+,2*ERRVEC
9204	032570					BR	7\$
9205	032570	032777	000400	146342	6\$::		
9206	032576	001421				BIT	#BIT08,\$SWR
9207	032600	005046				BEQ	2\$
9208	032602	117716	146332			CLR	-(SP)
9209	032606	001414				MOV	2\$,\$SWR,(SP)
9210	032610	022716	000111			BEQ	8\$
9211	032614	002411				CMP	#111,(SP)
						BLT	8\$

 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: CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
 ;: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,\$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 #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,\$SWR ;:LOOP ON SPEC. TEST?
 BEQ 2\$;:BR IF NO
 CLR -(SP) ;:CLEAR A TEMP. LOCATION
 MOV 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

9212	032616	011637	001102		MOV	(SP), \$STSTNM	UPDATE THE TEST NUMBER
9213	032622	005316			DEC	(SP)	BACKUP BY ONE
9214	032624	006316			ASL	(SP)	SCALE THE TEST NUMBER AS AN INDEX
9215	032626	062716	033032		ADD	\$\$SW08TBL, (SP)	FORM THE ADDRESS OF TEST POINTER
9216	032632	013637	001106		MOV	2(SP)+, \$LPADR	SET LOOP ADDRESS TO DESIRED TEST
9217	032636	000466			BR	\$OVER	GO LOOP ON THE TEST
9218	032640	005726		8\$:	TST	(SP)+	CLEAN THE BAD TEST NUMBER OFF OF THE STACK
9219	032642	105737	001103	2\$:	TSTB	\$ERFLG	HAS AN ERROR OCCURRED?
9220	032646	001421			BEQ	3\$	BR IF NO
9221	032650	123737	001115	001103	CMPB	\$ERMAX, \$ERFLG	MAX. ERRORS FOR THIS TEST OCCURRED?
9222	032656	101015			BHI	3\$	BR IF NO
9223	032660	032777	001000	146252	BIT	8BIT09, 2SWR	LOOP ON ERROR?
9224	032666	001404			BEQ	4\$	BR IF NO
9225	032670	013737	001110	001106	7\$:	MOV	\$LPERR, \$LPADR
9226	032676	000446			BR	\$OVER	SET LOOP ADDRESS TO LAST SCOPE
9227	032700	105037	001103		4\$:	CLRB	\$ERFLG
9228	032704	005037	001262		CLR	\$TIMES	ZERO THE ERROR FLAG
9229	032710	000415			BR	1\$	CLEAR THE NUMBER OF ITERATIONS TO MAKE
9230	032712	032777	004000	146220	3\$:	BIT	8BIT11, 2SWR
9231	032720	001011			BNE	1\$	ESCAPE TO THE NEXT TEST
9232	032722	005737	001304		TST	\$PASS	INHIBIT ITERATIONS?
9233	032726	001406			BEQ	1\$	BR IF YES
9234	032730	005237	001104		INC	\$ICNT	IF FIRST PASS OF PROGRAM
9235	032734	023737	001262	001104	CMP	\$TIMES, \$ICNT	INCREMENT ITERATION COUNT
9236	032742	002024			BGE	\$OVER	CHECK THE NUMBER OF ITERATIONS MADE
9237	032744	012737	000001	001104	1\$:	MOV	81, \$ICNT
9238	032752	013737	033030	001262	MOV	\$MXCNT, \$TIMES	BR IF MORE ITERATION REQUIRED
9239	032760	105237	001102		SSVLAD:	STSTNM	REINITIALIZE THE ITERATION COUNTER
9240	032764	113737	001102	001302	MOV	\$STSTNM, \$STSTNM	SET NUMBER OF ITERATIONS TO DO
9241	032772	011637	001106		MOV	(SP), \$LPADR	COUNT TEST NUMBERS
9242	032776	011637	001110		MOV	(SP), \$LPERR	SET TEST NUMBER IN APT MAILBOX
9243	033002	005037	001264		CLR	\$ESCAPE	SAVE SCOPE LOOP ADDRESS
9244	033006	112737	000001	001115	MOV	81, \$ERMAX	SAVE ERROR LOOP ADDRESS
9245	033014	013777	001102	146120	SOVER:	STSTNM, 2015PLAY	CLEAR THE ESCAPE FROM ERROR ADDRESS
9246	033022	013716	001106		MOV	\$LPADR, (SP)	ONLY ALLOW ONE(1) ERROR ON NEXT TEST
9247	033026	000002			RTI		DISPLAY TEST NUMBER
9248	033030	003720			\$MXCNT:	2000.	FUDGE RETURN ADDRESS
9249	033032				\$SW08TBL:		FIXES PS
9250	033032	003146			.WORD	TST1+2	MAX. NUMBER OF ITERATIONS
9251	033034	003264			.WORD	TST2+2	STARTING ADDRESS OF TEST 1
9252	033036	003352			.WORD	TST3+2	STARTING ADDRESS OF TEST 2
9253	033040	004470			.WORD	TST4+2	STARTING ADDRESS OF TEST 3
9254	033042	004572			.WORD	TST5+2	STARTING ADDRESS OF TEST 4
9255	033044	004750			.WORD	TST6+2	STARTING ADDRESS OF TEST 5
9256	033046	005076			.WORD	TST7+2	STARTING ADDRESS OF TEST 6
9257	033050	005202			.WORD	TST10+2	STARTING ADDRESS OF TEST 7
9258	033052	006126			.WORD	TST11+2	STARTING ADDRESS OF TEST 10
9259	033054	006244			.WORD	TST12+2	STARTING ADDRESS OF TEST 11
9260	033056	006354			.WORD	TST13+2	STARTING ADDRESS OF TEST 12
9261	033060	006616			.WORD	TST14+2	STARTING ADDRESS OF TEST 13
9262	033062	006750			.WORD	TST15+2	STARTING ADDRESS OF TEST 14
9263	033064	007160			.WORD	TST16+2	STARTING ADDRESS OF TEST 15
9264	033066	007350			.WORD	TST17+2	STARTING ADDRESS OF TEST 16
9265	033070	007472			.WORD	TST20+2	STARTING ADDRESS OF TEST 17
9266	033072	010262			.WORD	TST21+2	STARTING ADDRESS OF TEST 20
9267	033074	010660			.WORD	TST22+2	STARTING ADDRESS OF TEST 21
							STARTING ADDRESS OF TEST 22

E14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 172
SCOPE HANDLER ROUTINE

SEQ 0172

9268	033076	011072	.WORD	TST23+2	STARTING ADDRESS OF TEST 23
9269	033100	011310	.WORD	TST24+2	STARTING ADDRESS OF TEST 24
9270	033102	011434	.WORD	TST25+2	STARTING ADDRESS OF TEST 25
9271	033104	011566	.WORD	TST26+2	STARTING ADDRESS OF TEST 26
9272	033106	011744	.WORD	TST27+2	STARTING ADDRESS OF TEST 27
9273	033110	012076	.WORD	TST30+2	STARTING ADDRESS OF TEST 30
9274	033112	012372	.WORD	TST31+2	STARTING ADDRESS OF TEST 31
9275	033114	012566	.WORD	TST32+2	STARTING ADDRESS OF TEST 32
9276	033116	013022	.WORD	TST33+2	STARTING ADDRESS OF TEST 33
9277	033120	013210	.WORD	TST34+2	STARTING ADDRESS OF TEST 34
9278	033122	013442	.WORD	TST35+2	STARTING ADDRESS OF TEST 35
9279	033124	013630	.WORD	TST36+2	STARTING ADDRESS OF TEST 36
9280	033126	014052	.WORD	TST37+2	STARTING ADDRESS OF TEST 37
9281	033130	014240	.WORD	TST40+2	STARTING ADDRESS OF TEST 40
9282	033132	014432	.WORD	TST41+2	STARTING ADDRESS OF TEST 41
9283	033134	014626	.WORD	TST42+2	STARTING ADDRESS OF TEST 42
9284	033136	015032	.WORD	TST43+2	STARTING ADDRESS OF TEST 43
9285	033140	015236	.WORD	TST44+2	STARTING ADDRESS OF TEST 44
9286	033142	016102	.WORD	TST45+2	STARTING ADDRESS OF TEST 45
9287	033144	016306	.WORD	TST46+2	STARTING ADDRESS OF TEST 46
9288	033146	016512	.WORD	TST47+2	STARTING ADDRESS OF TEST 47
9289	033150	016730	.WORD	TST50+2	STARTING ADDRESS OF TEST 50
9290	033152	017134	.WORD	TST51+2	STARTING ADDRESS OF TEST 51
9291	033154	017340	.WORD	TST52+2	STARTING ADDRESS OF TEST 52
9292	033156	017464	.WORD	TST53+2	STARTING ADDRESS OF TEST 53
9293	033160	017610	.WORD	TST54+2	STARTING ADDRESS OF TEST 54
9294	033162	017734	.WORD	TST55+2	STARTING ADDRESS OF TEST 55
9295	033164	020060	.WORD	TST56+2	STARTING ADDRESS OF TEST 56
9296	033166	020204	.WORD	TST57+2	STARTING ADDRESS OF TEST 57
9297	033170	020250	.WORD	TST60+2	STARTING ADDRESS OF TEST 60
9298	033172	020322	.WORD	TST61+2	STARTING ADDRESS OF TEST 61
9299	033174	020600	.WORD	TST62+2	STARTING ADDRESS OF TEST 62
9300	033176	020762	.WORD	TST63+2	STARTING ADDRESS OF TEST 63
9301	033200	021374	.WORD	TST64+2	STARTING ADDRESS OF TEST 64
9302	033202	021562	.WORD	TST65+2	STARTING ADDRESS OF TEST 65
9303	033204	022266	.WORD	TST66+2	STARTING ADDRESS OF TEST 66
9304	033206	022476	.WORD	TST67+2	STARTING ADDRESS OF TEST 67
9305	033210	022602	.WORD	TST70+2	STARTING ADDRESS OF TEST 70
9306	033212	023224	.WORD	TST71+2	STARTING ADDRESS OF TEST 71
9307	033214	023646	.WORD	TST72+2	STARTING ADDRESS OF TEST 72
9308	033216	023756	.WORD	TST73+2	STARTING ADDRESS OF TEST 73
9309	033220	024610	.WORD	TST74+2	STARTING ADDRESS OF TEST 74
9310	033222	025266	.WORD	TST75+2	STARTING ADDRESS OF TEST 75
9311	033224	025502	.WORD	TST76+2	STARTING ADDRESS OF TEST 76
9312	033226	025760	.WORD	TST77+2	STARTING ADDRESS OF TEST 77
9313	033230	026040	.WORD	TST100+2	STARTING ADDRESS OF TEST 100
9314	033232	026232	.WORD	TST101+2	STARTING ADDRESS OF TEST 101
9315	033234	026456	.WORD	TST102+2	STARTING ADDRESS OF TEST 102
9316	033236	026604	.WORD	TST103+2	STARTING ADDRESS OF TEST 103
9317	033240	027126	.WORD	TST104+2	STARTING ADDRESS OF TEST 104
9318	033242	027660	.WORD	TST105+2	STARTING ADDRESS OF TEST 105
9319	033244	030062	.WORD	TST106+2	STARTING ADDRESS OF TEST 106
9320	033246	030302	.WORD	TST107+2	STARTING ADDRESS OF TEST 107
9321	033250	030516	.WORD	TST110+2	STARTING ADDRESS OF TEST 110
9322	033252	031214	.WORD	TST111+2	STARTING ADDRESS OF TEST 111
9323	033254	031772	.WORD	SEOP+2	ADDRESS OF END OF PASS

F14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 173
SCOPE HANDLER ROUTINE

SEQ 0173

```

9324 033256 043652 .WORD ABTFAIL+2 ;ADDRESS OF ABORT FAILURE HANDLER
9325
9326 .SBTTL APT COMMUNICATIONS ROUTINE
9327
9328 *****
9329 033260 112737 000001 033524 $ATY1: MOVB #1,$FFLG ;:TO REPORT FATAL ERROR
9330 033266 112737 000001 033522 $ATY3: MOVB #1,$MFLG ;:TO TYPE A MESSAGE
9331 033274 000403 BR $ATYC
9332 033276 112737 000001 033524 $ATY4: MOVB #1,$FFLG ;:TO ONLY REPORT FATAL ERROR
9333 033304 $ATYC:
9334 033304 010046 MOV RO,-(SP) ;:PUSH RO ON STACK
9335 033306 010146 MOV RI,-(SP) ;:PUSH RI ON STACK
9336 033310 105737 033527 TSTB $MFLG ;:SHOULD TYPE A MESSAGE?
9337 033314 001450 BEQ 5$ ;:IF NOT: BR
9338 033316 122737 000001 001316 CMPB #APTENV,$ENV ;:OPERATING UNDER APT?
9339 033324 001031 BNE 3$ ;:IF NOT: BR
9340 033326 132737 000100 001317 BITB #APTSPool,$ENV ;:SHOULD SPOOL MESSAGES?
9341 033334 001425 BEQ 3$ ;:IF NOT: BR
9342 033336 017600 000004 MOV #4(SP),RO ;:GET MESSAGE ADDR.
9343 033342 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDR.
9344 033350 005737 001276 1$: TST $MSGTYPE ;:SEE IF DONE W/ LAST XMISSION?
9345 033354 001375 BNE 1$ ;:IF NOT: WAIT
9346 033356 010037 001312 MOV RO,$MSGAD ;:PUT ADDR IN MAILBOX
9347 033362 105720 2$: TSTB (RO)+ ;:FIND END OF MESSAGE
9348 033364 001376 BNE 2$
9349 033366 163700 001312 SUB $MSGAD,RO ;:SUB START OF MESSAGE
9350 033372 006200 ASR RO ;:GET MESSAGE LGTH IN WORDS
9351 033374 010037 001314 MOV RO,$MSGGLT ;:PUT LENGTH IN MAILBOX
9352 033400 012737 000004 001276 MOV #4,$MSGTYPE ;:TELL APT TO TAKE MSG.
9353 033406 000413 BR 5$
9354 033410 017637 000004 033434 3$: MOV #4(SP),4$ ;:PUT MSG ADDR IN JSR LINKAGE
9355 033416 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDRESS
9356 033424 013746 MOV 177776,-(SP) ;:PUSH 177776 ON STACK
9357 033430 004737 043662 JSR PC,$TYPE ;:CALL TYPE MACRO
9358 033434 000000 4$: .WORD 0
9359 033436 5$:
9360 033436 105737 033524 10$: TSTB $FFLG ;:SHOULD REPORT FATAL ERROR?
9361 033442 001416 BEQ 12$ ;:IF NOT: BR
9362 033444 005737 001316 TST $ENV ;:RUNNING UNDER APT?
9363 033450 001413 BEQ 12$ ;:IF NOT: BR
9364 033452 005737 001276 11$: TST $MSGTYPE ;:FINISHED LAST MESSAGE?
9365 033456 001375 BNE 11$ ;:IF NOT: WAIT
9366 033460 017637 000004 001300 MOV #4(SP),$FATAL ;:GET ERROR #
9367 033466 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDR.
9368 033474 005237 001276 INC $MSGTYPE ;:TELL APT TO TAKE ERROR
9369 033500 105037 033524 12$: CLRB $FFLG ;:CLEAR FATAL FLAG
9370 033504 105037 033523 CLRB $LFLG ;:CLEAR LOG FLAG
9371 033510 105037 033522 CLRB $MFLG ;:CLEAR MESSAGE FLAG
9372 033514 012601 MOV (SP)+,R1 ;:POP STACK INTO R1
9373 033516 012600 MOV (SP)+,RO ;:POP STACK INTO RO
9374 033520 000207 RTS PC ;:RETURN
9375 033522 000 $MFLG: .BYTE 0 ;:MESSG. FLAG
9376 033523 000 $LFLG: .BYTE 0 ;:LOG FLAG
9377 033524 000 $FFLG: .BYTE 0 ;:FATAL FLAG
9378 033526 .EVEN
9379 000200 APTSIZE=200

```

G14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 174
APT COMMUNICATIONS ROUTINE

SEQ 0174

```

9380      000001      APTENV=001
9381      000100      APTSPool=100
9382      000040      APTCSUP=040
9383      .SBTTL      ERROR HANDLER ROUTINE
9384
9385      ;*****
9386      ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
9387      ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
9388      ;AND GO TO TYPERR ON ERROR
9389      ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
9390      ;SW15=1      HALT ON ERROR
9391      ;SW13=1      INHIBIT ERROR TYPEOUTS
9392      ;SW10=1      BELL ON ERROR
9393      ;SW09=1      LOOP ON ERROR
9394      ;CALL
9395      ;*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
9396
9397      033526      SERROR:
9398      033526      104407      CKSWR      ;TEST FOR CHANGE IN SOFT-SWR
9399      033530      105237      001103      7S:      INCB      SERFLG      ;SET THE ERROR FLAG
9400      033534      001775      BEQ      7S      ;DON'T LET THE FLAG GO TO ZERO
9401      033536      013777      001102      145376      MOV      STSTNM,2DISP      ;DISPLAY TEST NUMBER AND ERROR FLAG
9402      033544      032777      002000      145366      BIT      #BIT10,2SWR      ;BELL ON ERROR?
9403      033552      001402      BEQ      1S      ;NO - SKIP
9404      033554      104401      001266      TYPE      SBELL      ;RING BELL
9405      033560      005237      001112      1S:      INC      SERRTL      ;COUNT THE NUMBER OF ERRORS
9406      033564      011637      001116      MOV      (SP),SERRPC      ;GET ADDRESS OF ERROR INSTRUCTION
9407      033570      162737      000002      001116      SUB      #2,SERRPC
9408      033576      117737      145314      001114      MOV      2SERRPC,SITEMB      ;STRIP AND SAVE THE ERROR ITEM CODE
9409      033604      032777      020000      145326      BIT      #BIT13,2SWR      ;SKIP TYPEOUT IF SET
9410      033612      001055      BNE      20S      ;SKIP TYPEOUTS
9411      033614      021627      001002      CMP      (SP),#1002      ;IF RETURN PC LESS THAN 1002
9412      033620      101046      BHI      12S      ;ERROR IS ILLEGAL TRAP
9413      ;;PROCESS UNEXPECTED TRAP OR INTERRUPT
9414      033622      016637      000004      001116      MOV      4(SP),SERRPC      ;GET PC AT TIME OF FALSE TRAP
9415      033630      162737      000002      001116      SUB      #2,SERRPC      ;ADJUST PC
9416      033636      104401      033702      TYPE      10S      ;TYPE HEADER
9417      033642      013746      001116      MOV      SERRPC,-(SP)      ;SAVE SERRPC FOR TYPEOUT
9418      033646      104402      TYPE      TPOC      ;GO TYPE--OCTAL ASCII(ALL DIGITS)
9419      033650      104401      033710      TYPE      11S
9420      033654      162716      000004      SUB      #4,(SP)      ;GET FALSE TRAP VECTOR ADDR
9421      033660      011637      001116      MOV      (SP),SERRPC
9422      033664      013746      001116      MOV      SERRPC,-(SP)      ;SAVE SERRPC FOR TYPEOUT
9423      033670      104402      TYPE      TPOC      ;GO TYPE--OCTAL ASCII(ALL DIGITS)
9424      033672      104401      001273      TYPE      SCRLF
9425      033676      022626      CMP      (SP)+,(SP)+      ;POP FALSE TRAP VECTOR PC&ADDR
9426      033700      000422      BR      20S
9427      033702      050200      036503      000040      10S:      .ASCIZ      <200>'PC= '
9428      033710      020040      047125      054105      11S:      .ASCIZ      ' UNEXPECTED TRAP TO '
9429      033716      042520      052103      042105
9430      033724      052040      040522      020120
9431      033732      047524      000040
9432      .EVEN
9433      033736      12S:
9434      033736      004737      034050      JSR      PC,TYPERR      ;GO TO USER ERROR ROUTINE
9435      033742      104401      001273      TYPE      ,SCRLF

```

H14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 175
ERROR HANDLER ROUTINE

SEQ 0175

```

9436 033746 122737 000001 001316 20$: CMPB #APTENV,SENV ;: RUNNING IN APT MODE
9437 033746 001007 000001 001316 BNE 2$ ;: NO SKIP APT ERROR REPORT
9438 033754 001007 000001 001316 MOV B,2$ ;: SET ITEM NUMBER AS ERROR NUMBER
9439 033756 113737 001114 033770 JSR PC,SATY4 ;: REPORT FATAL ERROR TO APT
9440 033764 004737 033276 21$: .BYTE 0
9441 033770 000 0 .BYTE 0
9442 033771 000 0 22$: BR 22$ ;: APT ERROR LOOP
9443 033772 000777 145140 2$: TST 2SWR ;: HALT ON ERROR
9444 033774 005777 145140 BPL 3$ ;: SKIP IF CONTINUE
9445 034000 100002 000000 HALT ;: HALT ON ERROR!
9446 034002 000000 000000 CKSWR ;: TEST FOR CHANGE IN SOFT-SWR
9447 034004 104407 001000 145124 3$: BIT #BIT09,2SWR ;: LOOP ON ERROR SWITCH SET?
9448 034006 032777 001000 145124 BEQ 4$ ;: BR IF NO
9449 034014 001402 001110 001264 4$: MOV $LPERR,(SP) ;: FUDGE RETURN FOR LOOPING
9450 034016 013716 001110 001264 TST $ESCAPE ;: CHECK FOR AN ESCAPE ADDRESS
9451 034022 005737 001264 5$: BEQ 5$ ;: BR IF NONE
9452 034026 001402 001264 MOV $ESCAPE,(SP) ;: FUDGE RETURN ADDRESS FOR ESCAPE
9453 034030 013716 001264 5$: CMP #SENDAD,2#42 ;: ACT-11 AUTO-ACCEPT?
9454 034034 022737 032160 000042 BNE 6$ ;: BRANCH IF NO
9455 034034 001001 000000 HALT ;: YES
9456 034042 001001 000000 6$: RTI ;: RETURN
9457 034044 000000 ;: *****
9458 034046 000002 ;: SBTTL TYPE ERROR ROUTINE
9459 ;: *ENTRY JSR PC,TYPERR
9460 ;: *RETURN RTS PC
9461 ;: *
9462 ;: *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
9463 ;: *ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
9464 ;: *ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
9465 ;: *THE ERROR.
9466 ;: *****
9467 TYPERR: SAVREG
9468 MOV B,$STNM,$TESTN ;: GET TEST NUMBER FOR REPORT
9469 BIC #177400,$TESTN ;: CLEAR UNUSED BITS
9470 MOV B,$ITEMB,R0 ;: ENTER ERROR NUMBER
9471 BIC #177400,R0 ;: CLEAR UNUSED BITS
9472 DEC R0 ;: FORM INDEX FOR ERROR TABLE
9473 ASL R0
9474 ASL R0
9475 ASL R0
9476 1$: ADD #ERRTB,R0 ;: FORM ADDRESS OF ERROR ENTRY
9477 MOV (R0)+,2$ ;: GET EM POINTER
9478 BEQ 3$ ;: BRANCH IF THERE ISN'T ONE
9479 TYPE , $CRLF ;: TYPE CARRIAGE RETURN LINE FEED
9480 TYPE ;: TYPE ERROR MESSAGE (EM)
9481 2$: .WORD 0 ;: EM POINTER GOES HERE
9482 3$: MOV (R0)+,4$ ;: GET DH POINTER
9483 BEQ 5$ ;: BRANCH IF THERE ISN'T ONE
9484 TYPE , $CRLF ;: TYPE CR-LF
9485 TYPE ;: TYPE DATA HEADER
9486 4$: .WORD 0 ;: DH POINTER GOES HERE
9487 5$: MOV (R0)+,R1 ;: GET DT POINTER
9488 BEQ 20$ ;: BRANCH IF THERE ARE NONE
9489
9490
9491

```

9492	034152	005004		CLR	R4	;SET INDENT SWITCH
9493	034154	012000		MOV	(R0)+,R0	;GET DF POINTER
9494	034156	012002		MOV	(R0)+,R2	;STORE NUMBER OF DH'S
9495	034160	104401	001273	TYPE	,SCRLF	
9496	034164	112003		10\$:	MOV	(R0)+,R3
9497	034166	105720		TST	(R0)+	;GET & STORE NUMBER OF DATA WORDS
9498	034170	005703		TST	R3	;BUMP PAST FORMAT WORD
9499	034172	001416		BEQ	14\$	;TEST IF ANY DATA FOR THIS HEADER
9500	034174	005704		TST	R4	;NO - SKIP DATA PRINT
9501	034176	001004		BNE	12\$	;CHECK FOR INDENT
9502	034200	013146		11\$:	MOV	2(R1)+,-(SP)
9503	034202	104402		TYPOC		;YES, GO INDENT
9504	034204	005303		DEC	R3	;PUT FIRST DATA WORD ON STACK
9505	034206	001403		BEQ	13\$	;TYPE IT
9506	034210	104401	050443	12\$:	TYPE	,SPACE2
9507	034214	000771		BR	11\$	;MORE DATA WORDS
9508	034216	104401	001273	13\$:	TYPE	,SCRLF
9509	034222	005710		TST	(R0)	;NO-BRANCH
9510	034224	001401		BEQ	14\$	;TYPE SEPARATORS
9511	034226	005104		COM	R4	;LOOP
9512	034230	005302		14\$:	DEC	R2
9513	034232	003414		BLE	20\$	;TYPE <CR><LF>
9514	034234	012037	034254	15\$:	MOV	(R0)+,18\$
9515	034240	001751		BEQ	10\$	;CHECK IF NEXT HEADER AVAILABLE
9516	034242	005704		TST	R4	;NO, DO NOT CHANGE INDENT
9517	034244	001402		BEQ	17\$	;CHANGE INDENT
9518	034246	104401	050443	17\$:	TYPE	,SPACE2
9519	034252	104401		18\$:	TYPE	,DH
9520	034254	000000		.WORD	0	;MORE DH'S?
9521	034256	104401	001273	20\$:	RESREG	;NO-BRANCH
9522	034262	000740		INC	ERRCNT	;GET NEXT DH POINTER
9523	034264	104414		BIT	#SW12,#SWR	;IF 0 GET DATA
9524	034266	005237	001632	25\$:	BEQ	25\$
9525	034272	032777	144640	20\$:	RESREG	;INDENT?
9526	034300	001421		INC	ERRCNT	;NO, BRANCH
9527	034302	022737	000024 001632	25\$:	BEQ	25\$
9528	034310	103015		CMP	#20.,ERRCNT	;TYPE INDENT
9529	034312	104401	053215	BHIS	25\$	;TYPE DH
9530				TYPE	,ABORT	;DH POINTER GOES HERE
9531	034316	005737	000042	TST	42	
9532	034322	001407		BEQ	22\$	
9533	034324	012706	001100	MOV	#STACK,SP	;GET DATA
9534	034330	012737	000001 032016	MOV	#1,SEOPCT	;INCREMENT THE ERROR COUNT
9535	034336	000137	031770	JMP	SEOP	;CHECK IF SWITCH 12 SET
9536	034342	000000		22\$:	HALT	;NO, RETURN
9537	034344	000207		25\$:	RTS	;CHECK IF ERROR THRESHOLD EXCEEDED
9538				.SBTTL		;NO, RETURN
9539				;		;TYPE "PROGRAM ABORTED BECAUSE
9540				;		;ERROR THRESHOLD EXCEEDED"
9541				;		;CHECK IF CHAIN MODE
9542				;		;NO, HALT PROCESSOR
9543	034346	005237	001662	NEXTINT:	INC	INTSET
9544	034352	000002		RTI		;BUMP THE INTERRUPT COUNTER
9545				.SBTTL		
9546				;		
9547				;		

RK611 INTERRUPT HANDLER
MOST INTERRUPTS FROM THE RK611 ARE HANDLED BY THIS ROUTINE. ACTUAL

J14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 177
RK611 INTERRUPT HANDLER

SEQ 0177

```

9548      ;*      PROCESSING AS A RESULT OF THE INTERRUPT IS LEFT TO THE MAIN
9549      ;*      PROGRAM.  THE HANDLER JUST SETS A FLAG TO INDICATE THE
9550      ;*      INTERRUPT OCCURRED.
9551
9552      034354 000240      INTHLR: NOP
9553      034356 005237      INC      INTSET      ;BUMP THE INTERRUPT FLAG
9554      034362 J000002      RTI      ;RETURN.
9555
9556      .SBTTL MEMORY PARITY ERROR TRAP HANDLER
9557      ;*      MEMORY PARITY TRAPS WILL BE REPORTED BY THIS ROUTINE.  THE REPORT
9558      ;*      WILL INCLUDE THE PC AT TIME OF FAILURE AND ABORT THE PROGRAM.
9559
9560      034364 032777 020000 144546 PERHLR: BIT      #BIT13,2SWR      ;TEST IF INHIBIT REPORT
9561      034372 001003      BNE      1$      ;YES - SKIP
9562      034374 104401 053444      TYPE      ,EM3      ;TYPE PARITY ERROR MESSAGE
9563      034400 104402      TYPOC      ;AND PC VALUE
9564      034402 012737 000001 032016 1$: MOV      #1,SEOPCT      ;FORCE END OF PROGRAM
9565      034410 012706 001100      MOV      #STACK,SP      ;CLEAN OFF STACK
9566      034414 000137 031770      JMP      SEOP
9567
9568      .SBTTL LINE CLOCK INTERRUPT HANDLER
9569      ;*      THE LINE CLOCK INTERRUPT HANDLER WILL INCREMENT THE LCLKTK
9570      ;*      (LINE CLOCK TICK COUNTER) EACH TIME AN INTERRUPT OCCURS.
9571
9572      034420 005237 001674      LCKHLR: INC      LCLKTK      ;INCREMENT CLOCK TICK COUNTER
9573      034424 042777 000200 145256      BIC      #BIT7,2KWLADD      ;CLEAR MONITOR BIT
9574      034432 000002      RTI
9575
9576      ;*****
9577      .SBTTL OPTION PRESENT TEST AND SETUP
9578      ;*      THIS ROUTINE CHECKS IF THE MEMORY PARITY OPTION AND THE
9579      ;*      LINE CLOCK ARE ON THE SYSTEM.  FLAGS ARE SET IF PRESENT; CLEARED
9580      ;*      OTHERWISE, AND THE APPROPRIATE INTERRUPT VECTORS ARE SET UP.
9581      034434 104413      OPTTST: SAVREG
9582      034436 013746 000004      MOV      ERRVEC, -(SP)      ;STORE OLD NEM CONTENTS
9583      034442 013746 000006      MOV      ERRVEC+2, -(SP)
9584      034446 012737 034634 000004      MOV      #20$,ERRVEC      ;DET INTERRUPT FOR NEM
9585      034454 012737 000340 000006      MOV      #PR7,ERRVEC+2      ;SET PRIORITY
9586      034462 005037 001670      CLR      MEMPAR      ;CLEAR PARITY WORD FLAGS
9587      034466 042737 000100 001664      BIC      #PARPRE,OPTFLG      ;CLEAR PARITY PRESENT FLAG
9588      034474 005737 170200      TST      170200      ;TEST IF 11/70 REGISTER PRESENT
9589      034500 000240      NOP
9590      034502 000240      NOP
9591      034504 012737 177750 001672      MOV      #177750,CSRPTR      ;SET POINTER FOR 11/70
9592      034512 052737 002000 001664      BIS      #CP1170,OPTFLG      ;SET 11/70 FLAG
9593      034520 052737 000100 001664      BIS      #PARPRE,OPTFLG      ;SET PARITY PRESENT FLAG
9594      034526 000464      BR      35$      ;GO TO VECTOR SETUP
9595
9596      034530 013703 001714      3$: MOV      MEMBAS,R3      ;SET UP POINTER TO FIRST PARITY CSR
9597      034534 012704 000001      MOV      #1,R4      ;INIT MASK
9598      034540 012713 000001      6$: MOV      #1,(R3)      ;SET PARITY DETECT IN THAT CSR
9599      034544 005713      TST      (R3)
9600      034546 050437 001670      BIS      R4, MEMPAR      ;SET PARITY PRESENT BIT
9601      034552 032737 000100 001664      BIT      #PARPRE,OPTFLG      ;WAS PARITY PRESENT SET BEFORE
9602      034560 001017      BNE      10$      ;YES - SKIP
9603      034562 013700 001716      MOV      MMVECA,R0      ;SET UP FOR PARITY TRAPS

```

K14

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC - MACY11 27(1006) 31-JAN-77 16:34 PAGE 178
 DZR6KD.P11 31-JAN-77 16:26 OPTION PRESENT TEST AND SETUP

SEQ 0178

9604	034566	012720	034646		MOV	#40\$, (R0)+	; TO 40\$
9605	034572	012710	000340		MOV	#PR7, (R0)	
9606	034576	012700	063640		MOV	#0BUFF+224, R0	; SET POINTER TO BUFFER WHERE BAD PARITY
9607							IS USED IN THE TESTS
9608	034602	012713	000004		MOV	#BIT2, (R3)	; SET TO WRITE WRONG PARITY
9609	034606	012710	177777		MOV	#-1, (R0)	; WRITE WORD BAD
9610	034612	012713	000001		MOV	#1, (R3)	; SET TO DETECT PARITY ERROR
9611	034616	005710			TST	(R0)	; READ BAD WORD
9612	034620	062703	000002	10\$:	ADD	#2, R3	; BUMP TO NEXT CSR
9613	034624	000241			CLC		
9614	034626	006104			ROL	R4	; SHIFT MASK
9615	034630	001343			BNE	6\$	; TEST IF ALL DONE
9616	034632	000422			BR	35\$	; YES - SKIP
9617							
9618	034634	022626		20\$:	CMP	(SP)+, (SP)+	; CLEAR STACK
9619	034636	012737	034674	000004	MOV	#30\$, ERRVEC	; SET NEW NEM TRAP ADDRESS
9620	034644	000731			BR	3\$	; GO TO CSR CHECKS
9621							
9622	034646	022626		40\$:	CMP	(SP)+, (SP)+	; CLEAR STACK
9623	034650	010337	001672		MOV	R3, CSAPTR	; SET CSR POINTER FOR CSR TO BE USED
9624	034654	052737	000100	001664	BIS	#PARPRE, OPTFLG	; SET PARITY PRESENT FLAG
9625	034662	005013			CLR	(R3)	; CLEAR CSR
9626	034664	005010			CLR	(R0)	; CLEAR BAD WORD
9627	034666	012713	000001		MOV	#1, (R3)	; SET TO DETECT PARITY ERRORS
9628	034672	000752			BR	10\$	; CHECK NEXT CSR
9629							
9630	034674	022626		30\$:	CMP	(SP)+, (SP)+	; CLEAR STACK
9631	034676	000750			BR	10\$	; GO CHECK NEXT CSR
9632	034700	013700	001716		MOV	MMVECA, R0	; SET UP POINTER TO PARITY VECTOR
9633	034704	005737	001670		TST	MEMPAR	; TEST IF ANY PARITY PRESENT
9634	034710	001004			BNE	37\$	; YES - SKIP
9635	034712	032737	002000	001664	BIT	#CP1170, OPTFLG	; TEST IF 11/70
9636	034720	001405			BEQ	39\$	; NO - SKIP
9637	034722	012720	034364		MOV	#PERHLR, (R0)+	; SET UP PARITY VECTOR
9638	034726	012710	000340		MOV	#PR7, (R0)	; SET PRIORITY
9639	034732	000403			BR	38\$	
9640	034734	012720	000116		MOV	#116, (R0)+	; ELSE RESTORE TRAP CATCHER
9641	034740	005010			CLR	(R0)	
9642	034742	012737	034346	000004	MOV	#NEXINT, ERRVEC	; SET UP NEM VECTOR FOR LINE CLOCK TEST
9643	034750	005037	001662		CLR	INTSET	; CLEAR INTERRUPT COUNTER
9644	034754	013700	001712		MOV	KW1, VEC, R0	; SET POINTER TO VECTOR
9645	034760	012720	034420		MOV	#LCKHLR, (R0)+	; INSERT ADDRESS OF INTERRUPT HDLR
9646	034764	012710	000340		MOV	#PR7, (R0)	; INSERT PRIORITY
9647	034770	012777	000100	144712	MOV	#BIT6, #KW1LADD	; LOAD KW11-L FOR INTERRUPT ENABLE
9648	034776	000240			NOP		
9649	035000	000240			NOP		
9650	035002	000240			NOP		
9651	035004	005737	001662		TST	INTSET	; TEST IN NEM ON KW11-P REFERENCE
9652	035010	001003			BNE	4\$	; THIS BRANCH WILL BYPASS SET UP OF
9653							CLOCK OPTION
9654	035012	052737	100000	001664	BIS	#LCLKPR, OPTFLG	; SET CLOCK PRESENT FLAG
9655	035020	012701	000006		MOV	#6, R1	; RESTORE OLD VECTOR
9656	035024	005037	001662		CLR	INTSET	; CLEAR INT FLAG
9657	035030	012637	000006		MOV	(SP)+, ERRVEC+2	
9658	035034	012637	000004		MOV	(SP)+, ERRVEC	
9659	035040	012746	000000		MOV	#PRO, -(SP)	; RESTORE PRIORITY TO 0

```

9660 035044 012746 035052      MOV      #125,-(SP)
9661 035050 000002      RTI
9662 035052      12$:
9663 035052 104414      RESREG
9664 035054 000207      RTS      PC
9665
9666 ;*****
9667 .SBTTL LOOP ON INTERNAL ERROR
9668 ;* THIS ROUTINE IS USED TO PROVIDE TIGHT SCOPE LOOPS. THE CALLER
9669 ;* IS EXPECTED TO SET $LPERR TO THE START OF THE SCOPE LOOP
9670 ;* TO BE EXECUTED ON ERROR.
9671
9672 035056 032777 001000 144054 SCOP1$: BIT      #SW9,2SWR      ;CHECK IF LOOP ON ERROR
9673 035064 001405      BEQ      5$          ;NO, CONTINUE
9674 035066 105737 001103      TSTB     $ERFLG      ;CHECK IF ERROR OCCURRED
9675 035072 001402      BEQ      5$
9676 035074 013716 001110      MOV      $LPERR,(SP)    ;LOAD ERROR RETURN
9677 035100 000002      5$: RTI          ;RETURN
9678 .SBTTL LINE CLOCK CALIBRATE
9679 ;* WAITS FOR A LINE CLOCK INTERRUPT TO CALIBRATE THE INTERRUPTS
9680 ;* TO A MEANINGFUL TIME VALUE. IN ADDITION IT PRESETS
9681 ;* THE TIMCNT IF THERE IS NO LINE CLOCK. TIMCNT IS USED IN THE
9682 ;* LINE CLOCK SIMULATOR.
9683
9684 035102 005037 001674      CLKCAL: CLR      LCLKTK      ;CLEAR TICK COUNTER
9685 035106 032737 100000 001664 BIT      #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
9686 035114 001004      BNE      1$          ;YES - SKIP
9687 035116 012737 000020 001654 MOV      #16.,TIMCNT ;ELSE PRESET TIMCNT
9688 035124 000410      BR       2$          ;AND EXIT
9689 035126 005737 001662      1$: TST      INTSET      ;TEST IF INTERRUPT HAS OCCURRED
9690 035132 001005      BNE      2$          ;YES- ABORT CALIBRATION
9691 035134 005737 001674      TST      LCLKTK      ;WAIT FOR CLOCK TICK
9692 035140 001772      BEQ      1$          ;NOT YET - LOOP
9693 035142 005037 001674      CLR      LCLKTK      ;CLEAR TICK COUNT
9694 035146 000207      2$: RTS      PC          ;RETURN
9695
9696 .SBTTL LINE CLOCK SIMULATION ROUTINE
9697 ;* THIS ROUTINE IS USED TO SIMULATE THE LINE CLOCK. TO
9698 ;* DO THIS THE VALUE STORED IN MILCNT IS USED AS THE
9699 ;* BASE AND REPRESENTS THE NUMBER OF TIMES A DECREMENT
9700 ;* AND BRANCH LOOP CAN BE DONE IN 1 MILLISECOND. THE
9701 ;* TIMCNT VALUE IS DECREMENTED AND IF IT REACHED 0 THE
9702 ;* LINE CLOCK TICK COUNTER IS BUMPED. THEN THE TIMCNT
9703 ;* IS RESET TO 16 (REPRESENTS 16 MILLISECONDS PER LINE CLOCK
9704 ;* TICK).
9705 ;*
9706 ;* THUS THE ROUTINE RETURNS TO THE CALLER AFTER 1 MILLISECOND
9707 ;* AND BUMPS THE LINE CLOCK TICK COUNTER FOR EACH 16 CALLS.
9708 ;*
9709
9710 035150 010046      MYTIME: MOV      R0,-(SP)      ;SAVE R0
9711 035152 013700 001652      MOV      MILCNT,R0      ;SET COUNT
9712 035156 005737 001662      1$: TST      INTSET      ;TEST IF INTERRUPT SET
9713 035162 001012      BNE      2$          ;YES - SKIP
9714 035164 005300      DEC      R0          ;DECREMENT COUNT TO ZERO
9715 035166 001373      BNE      1$

```

M14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 180
LINE CLOCK SIMULATION ROUTINE

SEQ 0180

9716	035170	005337	001654		DEC	TIMCNT	; DECREMENT TIMCNT
9717	035174	001005			BNE	25	; IF NOT ZERO - EXIT
9718	035176	005237	001674		INC	LCLKTK	; ELSE BUMP TICK COUNTER
9719	035202	012737	000020	001654	MOV	#16, TIMCNT	; RESET TIME COUNT
9720	035210	012600			MOV	(SP)+, RO	; RESTORE RO
9721	035212	000207			RTS	PC	; RETURN
9722							
9723					.SBTTL	WAIT FOR INTERRUPT ROUTINE	
9724					;	THE ROUTINE IS ENTERED BY ONE OF FOURTEEN TRAP CALLS. THE CALL	
9725					;	SPECIFIES HOW MANY TICKS OF THE LINE CLOCK ARE TO ELAPSE	
9726					;	WHILE WAITING FOR INTERRUPT. IF INTERRUPT DOES NOT OCCUR	
9727					;	IN THAT PERIOD OF TIME, AN ERROR MESSAGE IS PREPARED	
9728					;	(BUT NOT PRINTED IN THE ROUTINE) AND THEN RETURNS TO THE	
9729					;	LOCATION FOLLOWING THE CALL. IF INTERRUPT OCCURS THE	
9730					;	RETURN IS BUMPED BY 2. NORMALLY AN ERROR CALL WILL	
9731					;	BE IN THE LOCATION AFTER THE CALL TO INTERRUPT WAIT.	
9732							
9733	035214	104413			IWAT8S:	SAVREG	; ENTRY FOR 8 SECOND WAIT
9734	035216	012700	000764		MOV	#500, RO	
9735	035222	000463			BR	WATSRT	
9736	035224	104413			IWAT1M:	SAVREG	; ENTRY FOR 1 MIN WAIT
9737	035226	012700	007246		MOV	#3750, RO	
9738	035232	000457			BR	WATSRT	
9739	035234	104413			IWAT2S:	SAVREG	; ENTRY FOR 2 SECOND WAIT
9740	035236	012700	000200		MOV	#128, RO	
9741	035242	000453			BR	WATSRT	
9742	035244	104413			IWAT1S:	SAVREG	; ENTRY FOR 1 SECOND WAIT
9743	035246	012700	000077		MOV	#63, RO	
9744	035252	000447			BR	WATSRT	
9745	035254	104413			IWAT159:	SAVREG	; ENTRY FOR 160 MS WAIT
9746	035256	012700	000012		MOV	#10, RO	
9747	035262	000443			BR	WATSRT	
9748	035264	104413			IWAT144:	SAVREG	; ENTRY FOR 144 MS WAIT
9749	035266	012700	000011		MOV	#9, RO	
9750	035272	000437			BR	WATSRT	
9751	035274	104413			IWAT128:	SAVREG	; ENTRY FOR 128 MS WAIT
9752	035276	012700	000010		MOV	#8, RO	
9753	035302	000433			BR	WATSRT	
9754	035304	104413			IWAT112:	SAVREG	; ENTRY FOR 112 MS WAIT
9755	035306	012700	000007		MOV	#7, RO	
9756	035312	000427			BR	WATSRT	
9757	035314	104413			IWAT96:	SAVREG	; ENTRY FOR 96 MS WAIT
9758	035316	012700	000006		MOV	#6, RO	
9759	035322	000423			BR	WATSRT	
9760	035324	104413			IWAT80:	SAVREG	; ENTRY FOR 80 MS WAIT
9761	035326	012700	000005		MOV	#5, RO	
9762	035332	000417			BR	WATSRT	
9763	035334	104413			IWAT64:	SAVREG	; ENTRY FOR 64 MS WAIT
9764	035336	012700	000004		MOV	#4, RO	
9765	035342	000413			BR	WATSRT	
9766	035344	104413			IWAT48:	SAVREG	; ENTRY FOR 48 MS WAIT
9767	035346	012700	000003		MOV	#3, RO	
9768	035352	000407			BR	WATSRT	
9769	035354	104413			IWAT32:	SAVREG	; ENTRY FOR 32 MS WAIT
9770	035356	012700	000002		MOV	#2, RO	
9771	035362	000403			BR	WATSRT	

N14

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 181
WAIT FOR INTERRUPT ROUTINE

SEQ 0181

```

9772 035364 104413          IWAT16: SAVREG          ;ENTRY FOR 16 MS WAIT
9773 035366 012700 000001    MOV      #1,RO
9774 035372 012746 000000    WATSRT: MOV      #PR0,-(SP)    ;LOAD PRIORITY 0 ON STACK
9775 035376 012746 035404    MOV      #SS,-(SP)    ;LOAD ADDRESS
9776 035402 000002    RTI
9777
9778 035404 012737 000020 001654 5$: MOV      #16,TIMCNT    ;PRESET TIME COUNTER
9779 035412 004737 035102    JSR      PC,CLKCAL    ;GO CALIBRATE THE CLOCK
9780 035416 005737 001662    1$: TST      INTSET    ;TEST IF INTERRUPT OCCURRED
9781 035422 001036          BNE      3$    ;YES - EXIT
9782 035424 032737 100000 001664 BIT      #LCLKPR,OPTFLG    ;TEST IF KW11-L AVAILABLE
9783 035432 001002          BNE      2$    ;YES - SKIP
9784 035434 004737 035150    JSR      PC,MYTIME    ;ELSE CALL SIMULATOR
9785 035440 023700 001674    2$: CMP      LCLKTK,RO    ;TEST IF ENOUGH TICKS COUNTED
9786 035444 103764          BLO      1$    ;NO - LOOP
9787 035446 104420          TGETRK    ;ELSE GET '611 REGS
9788 035450 013701 001540    MOV      T.CS1,R1    ;PUT CS1 IN R1- STRIP ALL BUT
9789 035454 042701 177741    BIC      #177741,R1    ;COMMAND CODE; INDEX INTO TABLE
9790 035460 016137 047600 001372 MOV      CMNDLB(R1),DH2N    ;AND SELECT HEADER TO IDENTIFY OPERATION
9791 035466 012737 053514 001370 MOV      #EM4,EM2N    ;MESSAGE (NO INTERRUPT OR INTERRUPT LATE)
9792 035474 013700 001302    MOV      $TESTN,RO    ;GET NUMBER OF PRESENT TEST
9793 035500 006300          ASL      RO    ;SHIFT FOR INDEX
9794 035502 016037 033032 001264 MOV      $SW08TB(RO), $ESCAPE    ;LOAD ESCAPE TO ABORT TESTS
9795 035510 162737 000002 001264 SUB      #2,$ESCAPE    ;BUT GO TO NEXT SCOPE CALL
9796 035516 000402          BR      4$
9797 035520 062716 000002    3$: ADD      #2,(SP)    ;BUMP RETURN AROUND ERROR
9798 035524 104414          4$: RESREG    ;RESTORE REGS
9799 035526 000002          RTI    ;RETURN
9800
9801          .SBTTL    "L" REGISTER LOADING ROUTINE
9802          ;*
9803          ;* THE PARAMETERS FOLLOWING THE CALL ARE TRANSFERRED INTO
9804          ;* THE "L" REGISTERS L.CS1-L.DCYL. L.MR1 IS NOT
9805          ;* LOADED IN THIS MANNER SINCE IT IS NOT COMMONLY LOADED
9806          ;* FOR AN OPERATION. L.CS2 IS LOADED FROM DRVNUM.
9807          ;*
9808          ;* CALL: JSR      R4,LRLoad
9809          ;* COMMAND
9810          ;* WORD COUNT
9811          ;* BUS ADDRESS
9812          ;* .BYTE          SECTOR ADDRESS
9813          ;* .BYTE          TRACK ADDRESS
9814          ;* CYLINDER ADDRESS
9815
9815 035530          LRLoad:
9816 035530 010046          MOV      RO,-(SP)    ;PUSH RO ON STACK
9817 035532 010146          MOV      R1,-(SP)    ;PUSH R1 ON STACK
9818 035534 012701 001600    MOV      #L.CS1,R1    ;GET ADDRESS OF L REGS
9819 035540 012700 000004    MOV      #4,RO    ;PRESET COUNT
9820 035544 012421    1$: MOV      (R4)+,(R1)+    ;MOVE FIRST FOUR WORDS INTO "L" REGS
9821 035546 005300          DEC      RO    ;CS1, WC, BA, DA
9822 035550 001375          BNE      1$
9823 035552 013721 001626    MOV      DRVNUM,(R1)+    ;LOAD DRIVE NUMBER
9824 035556 005721          TST      (R1)+    ;BUMP PAST ASOF
9825 035560 012411          MOV      (R4)+,(R1)    ;LOAD DCYL
9826 035562 012601          MOV      (SP)+,R1    ;POP STACK INTO R1
9827 035564 012600          MOV      (SP)+,RO    ;POP STACK INTO RO

```

B15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 182
"L" REGISTER LOADING ROUTINE

SEQ 0182

```

9828 035566 000204          RTS      R4
9829
9830
9831
9832
9833
9834
9835 035570 005037 001662   .SBTTL  LOAD RK611 FOR OPERATION
9836 035574 010046          ;*      THE REGISTER SETUP STORAGE IS TRANSFERRED TO THE RK611 REGISTER.
9837 035576 010146          ;*      THIS IS A STRAIGHT TRANSFER WITH NO CHECKING OR MANIPULATION
9838 035600 012700 001602   ;*      OF THE REGISTER CONTENTS. L.CS1 IS TRANSFERRED LAST AS IT
9839 035604 010201          ;*      SHOULD BE IF THE GO BIT IS SET.
9840 035606 062701 000002   LOADRK: CLR      INTSET      ;CLEAR INTERRUPT FLAG.
9841 035612 012021          MOV      R0,-(SP)    ;STORE REGISTER
9842 035614 012021          MOV      R1,-(SP)
9843 035616 012021          MOV      #L.WC,R0    ;GET ADDRESS OF SETUP STORAGE WC
9844 035620 012011          MOV      R2,R1      ;GET BASE ADDRESS
9845 035622 062701 000006   ADD      #2,R1      ;PUT R1 PAST RKCS1
9846 035626 012021          MOV      (R0)+,(R1)+ ;LOAD WORD COUNT
9847 035630 012021          MOV      (R0)+,(R1)+ ;LOAD BUS ADDRESS
9848 035632 062701 000004   MOV      (R0)+,(R1)+ ;LOAD DISK ADDRESS
9849 035636 011011          MOV      (R0)+,(R1)+ ;LOAD CS2
9850 035640 013712 001600   ADD      #6,R1      ;BUMP R1 TO ASOF
9851 035644 012601          MOV      (R0)+,(R1)+ ;LOAD OFFSET
9852 035646 012600          MOV      (R0)+,(R1)+ ;LOAD CYLINDER
9853 035650 000002          ADD      #4,R1      ;BUMP R1 TO MR1
9854
9855          MOV      (R0),(R1)    ;LOAD MR1
9856          MOV      L.CS1,(R2)   ;LOAD CS1
9857          MOV      (SP)+,R1     ;RESTORE REGISTER
9858          MOV      (SP)+,R0     ;RESTORE REGISTER
9859          RTI                  ;RETURN
9860
9861
9862
9863
9864
9865
9866
9867
9868
9869
9870
9871
9872
9873
9874
9875
9876
9877
9878
9879
9880
9881
9882
9883

.SBTTL  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.
GETRK:  MOV      R0,-(SP)        ;STORE REGISTERS TO BE USED
        MOV      R1,-(SP)
        MOV      R3,-(SP)
        MOV      #T.CS1,R0      ;SET POINTER TO TEST TABLE
        MOV      R2,R1          ;SET POINTER TO RK611 BASE
        MOV      #10,R3         ;SET COUNT FOR 1ST 10 REGS
1$:     MOV      (R1)+,(R0)+     ;STORE RKCS1 THROUGH RKSPAR
        DEC      R3
        BNE     1$
        ADD      #2,R1          ;BUMP POINTER PAST RKDB
        TST      (R0)+          ;BUMP POINTER PAST T.RKDB
        MOV      #4,R3         ;SET COUNT FOR LAST 5 REGS
2$:     MOV      (R1)+,(R0)+     ;STORE RKMR1 THROUTH RKMR3
        DEC      R3
        BNE     2$
        MOV      (SP)+,R3       ;RESTORE REGISTERS
        MOV      (SP)+,R1
        MOV      (SP)+,R0
        RTI                    ;RETURN
.SBTTL  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.

```

```

9884 035730 016637 000002 001256 BITCNT: MOV 2(SP),STMP16 ;GET WORD WHOSE BITS ARE TO BE COUNTED
9885 035736 010346 MOV R3,-(SP) ;STORE R3
9886 035740 005037 001260 CLR STMP17 ;CLEAR STMP16 FOR COUNTING
9887 035744 012703 000021 MOV #17,R3 ;SET A SHIFT COUNTER
9888 035750 000241 CLC ;CLEAR CARRY
9889 035752 006037 001256 1$: ROR STMP16 ;ROTATE WORD
9890 035756 103407 BCS 3$ ;WAS BIT SHIFTED OUT A 1?
9891 035760 005303 2$: DEC R3 ;NO - DEC COUNT
9892 035762 001373 BNE 1$ ;LOOP IF NOT ZERO
9893 035764 012603 MOV (SP)+,R3 ;RESTORE R3
9894 035766 013766 001260 000002 MOV STMP17,2(SP) ;PUT COUNT OF BITS ON STACK
9895 035774 000204 RTS R4 ;RETURN
9896 035776 005237 001260 3$: INC STMP17 ;BUMP COUNT
9897 036002 000766 BR 2$ ;LOOP
9898
9899 .SBTTL MAINTENANCE CLOCK ROUTINE
9900 ;*
9901 ;* THE PARAMETERS PASSED TO THIS ROUTINE ARE LOCATED IN THE
9902 ;* ADDRESS AFTER THE CALL. THE FIRST BYTE CONTAINS THE NUMBER
9903 ;* OF PHASES ADDRESSES THE CALLING ROUTINE WANTS THE CONTROLLER
9904 ;* Clocked THROUGH AND THE SECOND BYTE CONTAINS THE NUMBER OF
9905 ;* CLOCK TRANSITIONS(PARTIAL PHASES) TO BE DONE.
9906
9907 036004 010046 MCLOCK: MOV R0,-(SP) ;PUSH R0 ON STACK
9908 036006 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
9909 036010 112400 MOVB (R4)+,R0 ;GET NUMBER OF CONTROLLER PHASE ADDRESSES
9910 036012 112401 MOVB (R4)+,R1 ;GET PARTIAL PHASE ADDRESS COUNT
9911
9912 036014 006300 ASL R0 ;MULTIPLY PHASE ADDRESS COUNT BY 4
9913 036016 006300 ASL R0
9914 036020 060100 ADD R1,R0 ;ADD IN PARTIALS
9915 036022 052762 000400 000026 1$: BIS #MCLK,RKMR1(R2) ;SET CLOCK
9916 036030 042762 000400 000026 BIC #MCLK,RKMR1(R2) ;CLEAR MCLK
9917 036036 005300 DEC R0 ;DECREMENT COUNT
9918 036040 001370 BNE 1$ ;LOOP IF NOT ZERO
9919 036042 012601 MOV (SP)+,R1 ;POP STACK INTO R1
9920 036044 012600 MOV (SP)+,R0 ;POP STACK INTO R0
9921 036046 000204 RTS R4
9922
9923 .SBTTL READ AND SORT HEADERS
9924 ;*
9925 ;* THE HEADERS IN THE CYLINDER AND TRACK SPECIFIED BY
9926 ;* THE FIELDS IN THE "L" REGISTERS ARE READ AND STORED IN
9927 ;* ASCENDING ORDER. CONTROLLER ERRORS ARE CHECKED IN THE
9928 ;* READ HEADER OPERATION AND DATA LATE IS CHECKED AFTER
9929 ;* EACH READ OF THE DATA BUFFER.
9930
9931 CALL: JSR R4,RDSTHD
9932 TCHKOP ;RETURN POINT IF CERR IN READ HDR
9933 ERROR 4 ;OR 5, 6, 7
9934 ERROR 13 ;RETURN IF DATA LATE IN DB UNLOAD
9935 ERROR 2 ;RETURN IF TO SLOW OR
;IF HDR 0 NOT FOUND
9936
9937 036050 104413 RDSTHD: SAVREG
9938 036052 032737 100000 001664 BIT #LCLKPR,OPTFLG ;TEST IF CLOCK PRESENT
9939 036060 001402 BEQ 20$ ;NO - SKIP
9940 036062 005077 143622 CLR #KWLADD ;RESET INTERRUPT

```

9940	036066	012700	000026		20\$:	MOV	#26,R0	;PRESET FOR 26 SECTOR FORMAT
9941	036072	032737	010000	001600		BIT	#CFAT,L.CS1	;IS 24 SECTOR MODE SET?
9942	036100	001402				BEQ	1\$	;NO - SKIP
9943	036102	012700	000024			MOV	#24,R0	;ELSE CHANGE TO 24 SECTOR MODE
9944	036106	012701	061414		1\$:	MOV	#IBUFF,R1	;SET POINTER TO INPUT BUFFER
9945	036112	010005				MOV	R0,R5	;SAVE NUMBER OF SECTORS
9946	036114	010104				MOV	R1,R4	;SAVE IBUFF ADDRESS
9947	036116	010203				MOV	R2,R3	;SET UP POINTER TO RKDB
9948	036120	062703	000024			ADD	#RKDB,R3	
9949	036124	013762	001626	000010		MOV	DRVNUM,RKCS2(R2)	;LOAD DRIVE NUM
9950	036132	013762	001614	000020		MOV	L.DCYL,RKDCYL(R2)	;LOAD CYLINDER NUM
9951	036140	013762	001606	000006		MOV	L.DA,RKDA(R2)	;LOAD TRACK AND SECTOR
9952								
9953	036146	012737	000020	001654	2\$:	MOV	#16,TIMCNT	;SET TIME COUNTER
9954	036154	005037	001662			CLR	INTSET	;CLEAR INTERRUPT FLAG
9955	036160	005037	001674			CLR	LCLKTK	;CLEAR TICK COUNTER
9956	036164	013762	001600	000000		MOV	L.CS1,RKCS1(R2)	;LOAD COMMAND
9957								
9958	036172	005737	001662		3\$:	TST	INTSET	;TEST IF INT OCCURRED
9959	036176	001020				BNE	4\$	;YES - SKIP
9960	036200	004737	035150			JSR	PC,MYTIME	;WAIT 1 MS
9961	036204	005737	001674			TST	LCLKTK	;HAVE WE WAITED 16 MS?
9962	036210	001770				BEQ	3\$	;NO - LOOP ON WAIT
9963								
9964	036212	062766	000006	000006		ADD	#6,6(SP)	;SET RETURN FOR TO SLOW
9965	036220	104420				TGETRK		;GET RK REGS
9966	036222	012737	053514	001370		MOV	#EM4,EM2N	;LOAD MESSAGE "TO SLOW/NOT COMPLETE"
9967	036230	012737	050004	001372		MOV	#OPER24,DH2N	;LOAD COMMAND "READ HEADER" FOR REPORT
9968	036236	000466				BR	10\$	;SKIP
9969								
9970	036240	005762	000000		4\$:	TST	RKCS1(R2)	;TEST FOR CONTROLLER ERROR
9971	036244	100474				BMI	11\$	;YES - SKIP
9972								
9973	036246	011324				MOV	(R3),(R4)+	;STORE HEADERS
9974	036250	011324				MOV	(R3),(R4)+	
9975	036252	011324				MOV	(R3),(R4)+	
9976								
9977	036254	005762	000010			TST	RKCS2(R2)	;TEST IF DATA LATE
9978	036260	100443				BMI	8\$	;YES - SKIP
9979								
9980	036262	005300				DEC	R0	;DEC SECTOR COUNT
9981	036264	001330				BNE	2\$	;IF NOT ZERO - LOOP
9982								
9983	036266	032737	100000	001664		BIT	#LCLKPR,OPTFLG	;TEST IF CLOCK PRESENT
9984	036274	001403				BEQ	5\$	;NO - SKIP
9985	036276	012777	000100	143404		MOV	#BIT6,#KWLADD	;SET INTERRUPT ENABLE
9986	036304	032761	000037	000002	5\$:	BIT	#37,2(R1)	;HEADER AT TOP OF BUFF=HEAD 0?
9987	036312	001413				BEQ	6\$	;YES - SKIP
9988	036314	012124				MOV	(R1)+,(R4)+	;ELSE MOV THIS HEADER TO BOTTOM
9989	036316	012124				MOV	(R1)+,(R4)+	
9990	036320	012124				MOV	(R1)+,(R4)+	
9991								
9992	036322	005305				DEC	R5	;TEST FO INSURE HEAD 0 IS FOUND
9993	036324	001367				BNE	5\$	;IF ALL HEADERS NOT CHECKED - LOOP
9994	036326	012737	056264	001370		MOV	#EM56,EM2N	;ELSE "HEADER 0 NOT FOUND" MESSAGE
9995	036334	005037	001372			CLR	DH2N	

E15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 185
READ AND SORT HEADERS

SEQ 0185

```

9996 036340 000421          BR      9$          ;SKIP
9997
9998 036342 012700 061414  6$:  MOV      #IBUFF,R0          ;GET TOP OF IBUFF
9999 036346 012120          7$:  MOV      (R1)+,(R0)+        ;MOVE HEADERS SO THEY START AT TOP OF IBUFF
10000 036350 012120          MOV      (R1)+,(R0)+
10001 036352 012120          MOV      (R1)+,(R0)+
10002 036354 020004          CMP      R0,R4          ;ALL HEADERS MOVED?
10003 036356 001373          BNE      7$          ;NO - LOOP
10004
10005 036360 062766 000010 000006  ADD      #10,6(SP)        ;SET UP FOR GOOD RETURN
10006 036366 000423          BR      11$
10007
10008 036370 012737 054527 001500 8$:  MOV      #ENDLT,EM13N        ;"DATA LATE SET RESULT OF READ DB"
10009 036376 012737 056675 061010  MOV      #EMRDB,DF011A
10010 036404 062766 000004 000006  9$:  ADD      #4,6(SP)          ;SET ERROR RETURN
10011 036412 104420          TGETRK          ;GET RK REGS
10012 036414 013700 001302  10$:  MOV      $TESTN,R0        ;GET TEST NUMBER
10013 036420 006300          ASL      R0          ;SHIFT FOR INDEX
10014 036422 016037 033032 001264  MOV      $SW08TB(R0),$ESCAPE ;SET ESCAPE
10015 036430 162737 000002 001264  SUB      #2,$ESCAPE        ;TO NEXT SCOPE CALL
10016
10017 036436 104414          11$:  RESREG
10018 036440 000204          RTS      R4
10019
10020          .SBTTL  GET DRIVE STATUS
10021          ;*
10022          ;* THIS ROUTINE GETS ALL THE DRIVE STATUS AND PLACES IT IN $REG10
10023          ;* THROUGH $REG17. THESE REGISTORS ARE FIRST CLEARED TO ALL ONES AND
10024          ;* THEN IF ERROR OCCURS WHILE GETTING STATUS, THE 1'S ARE LEFT
10025          ;* IN THE REGISTERS.
10026          ;*
10027          ;*CALL: JSR      R4,GETDRS
10028          ;*      BR      ERROR PROCESSING          ERROR RETURN
10029          ;*      BR      NO ERROR PROCESSING          GOOD RETURN
10030
10031 036442 104413          GETDRS: SAVREG
10032 036444 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR ANY OLD ERRORS LAYING AROUND
10033 036452 012700 001202          MOV      #$REG10,R0        ;PRESET ALL STATUS STORAGE TO
10034 036456 012701 000010          MOV      #8,R1          ;ALL ONES
10035 036462 012720 177777  1$:  MOV      #177777,(R0)+
10036 036466 005301          DEC      R1
10037 036470 001374          BNE      1$
10038 036472 012700 001206          MOV      #$REG12,R0        ;SET POINTER TO REG12 FOR A01 & B01
10039 036476 012701 000001          MOV      #1,R1          ;PRESET FOR PAIR ONE.
10040 036502 005037 001230          CLR      $MP3          ;CLEAR ERROR SWITCH
10041 036506 013762 001610 000010  2$:  MOV      L,CS2,RKCS2(R2) ;LOAD DRIVE #
10042 036514 010162 000026          MOV      R1,RKMR1(R2) ;LOAD MR1
10043 036520 012762 000001 000000  MOV      #BIT0,RKCS1(R2) ;DO SELECT
10044 036526 012703 000050          MOV      #40.,R3        ;WAIT FOR A FEW MICRO RECORDS TO
10045 036532 005303          3$:  DEC      R3          ;BIT SELECT FINISH
10046 036534 001376          BNE      3$
10047 036536 032762 100000 000000  BIT      #CERR,RKCS1(R2) ;ANY ERROR SET AS A RESULT OF SELECT?
10048 036544 001415          BEQ      4$          ;NO - SKIP
10049 036546 032762 024000 000000  BIT      #CTO!SPAR,RKCS1(R2) ;CHECK IF TIMEOUT OR PARITY ERROR
10050 036554 001004          BNE      8$          ;YES - SKIP
10051 036556 032762 037400 000010  BIT      #37400,RKCS2(R2) ;TEST FOR ERRORS:
;      NED!UPE!MDS!UFE!NEM!PGE

```

F15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 186
GET DRIVE STATUS

SEQ 0186

10052	036564	001405				BEQ	4\$		NO - SKIP
10053	036566	012737	000001	001230	8\$:	MOV	#1, \$TMP3		SET ERROR FLAG
10054	036574	022020				CMP	(R0)+, (R0)+		BUMP TO LET THAT PAIR STAY ALL 1'S.
10055	036576	000404				BR	5\$		SKIP
10056	036600	016220	000034		4\$:	MOV	RKMR2(R2), (R0)+		STORE A WORD
10057	036604	016220	000036			MOV	RKMR3(R2), (R0)+		STORE B WORD
10058	036610	012762	100000	000000	5\$:	MOV	#CCLR, RKCS1(R2)		CLEAR ANY OLD ERROR IN CONTROLLER
10059	036616	005701				TST	R1		IS R1 A 0 (LAST TRANSFER, PAIR 0)
10060	036620	001410				BEQ	6\$		YES - SKIP
10061	036622	005201				INC	R1		ELSE BUMP TO NEXT PAIR
10062	036624	022701	000004			CMP	#4, R1		PAIR 3 JUST STORED?
10063	036630	001326				BNE	2\$		NO - SKIP
10064	036632	005001				CLR	R1		ELSE SET TO PAIR 0
10065	036634	012700	001202			MOV	#SREG10, R0		PRESET POINTER FOR PAIR 0
10066	036640	000722				BR	2\$		GO GET THEM
10067	036642	104414			6\$:	RESREG			EXIT HERE
10068	036644	005737	001230			TST	\$TMP3		ANY ERROR IN STATUS GETTING
10069	036650	001001				BNE	7\$		YES - SKIP
10070	036652	005724				TST	(R4)+		ELSE BUMP PART ERROR
10071	036654	000204			7\$:	RTS	R4		RETURN
10072									
10073					.SBTTL				SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST
10074					*				THE SUBSYSTEM IS INITIALIZED WITH A SUBSYSTEM CLEAR
10075					*				COMMAND. CERR AND DI ARE MONITORED FOR A SHORT
10076					*				PERIOD OF TIME DURING WHICH THEY SHOULD BOTH RESET.
10077					*				
10078					*				IF THEY DO RESET, READY IS TESTED TO INSURE IF SETS.
10079					*				
10080					*				IF ANY OF THESE THREE CONDITIONS ARE NOT MET AN APPROPRIATE
10081					*				ERROR MESSAGE IS PREPARED AND REPORTED WHEN THE ROUTINE
10082					*				RETURN TO THE CALL. IF EVERY THING IS GOOD, THE RETURN
10083					*				SKIPS OVER THE ERROR CALL AND TEST ABORT.
10084					*				
10085					*				THE USUAL CALL TO THIS ROUTINE WILL BE FOLLOWED BY
10086					*				AN ERROR MESSAGE AND BRANCH TO END OF TEST. THIS
10087					*				IS DONE BECAUSE FAILURE TO INITIALIZE CORRECTLY IS FATAL TO
10088					*				THE TEST.
10089					*				
10090	036656				SSINIT:				
10091	036656	010046				MOV	R0, -(SP)		PUSH R0 ON STACK
10092	036660	010146				MOV	R1, -(SP)		PUSH R1 ON STACK
10093	036662	012701	000007			MOV	#7, R1		SET CLEAR COUNT
10094	036666	012700	001600			MOV	#L, CS1, R0		GET ADDRESS OF "L" REGS
10095	036672	012720	000100			MOV	#100, (R0)+		PRESET CS1
10096	036676	005020			7\$:	CLR	(R0)+		CLEAR THE NEXT
10097	036700	005301				DEC	R1		COUNT 0?
10098	036702	001375				BNE	7\$		NO - LOOP
10099	036704	012762	000040	000010		MOV	#SCLR, RKCS2(R2)		CLEAR SUBSYSTEM
10100	036712	012737	000012	001222		MOV	#10, \$TMP0		SET A COUNTER
10101	036720	016237	000000	001540	1\$:	MOV	RKCS1(R2), T.CS1		GET CS1
10102	036726	032737	140000	001540		BIT	#CERR!DI, T.CS1		TEST IF ERROR OR DI SET
10103	036734	001433				BEQ	2\$		NO - SKIP TO READY TEST
10104	036736	005337	001222			DEC	\$TMP0		ELSE DECREMENT COUNTER
10105	036742	001366				BNE	1\$		AND LOOP
10106	036744	032737	100000	001540		BIT	#CERR, T.CS1		TEST - IS IT CERR STILL SET
10107	036752	001404				BEQ	3\$		NO - SKIP TO DI MESSAGE

G15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 187
SUBSYSTEM INITIALIZE AND INITIALIZE STATE TEST

SEQ 0187

10108	036754	012737	053607	001400		MOV	#EM5,EM3N	;MESSAGE (SUBSYS CLR NOT RESET ERROR)
10109	036754	000403				BR	4\$	
10110	036764	012737	053653	001400	3\$:	MOV	#EM6,EM3N	;MESSAGE (SUBSYS CLEAR NOT RESET DI)
10111	036772	104420			4\$:	TGETRK		
10112	036774	013700	001302			MOV	\$TESTN,RO	;GET PRESENT TEST NUMBER
10113	037000	006300				ASL	RO	;SHIFT FOR INDEX
10114	037002	016037	033032	001264		MOV	\$SWOBTBL(RO), \$ESCAPE	;LOAD ESCAPE TO ABORT TEST
10115	037010	162737	000002	001264		SUB	#2, \$ESCAPE	;SET TO NEXT SCOPE CALL
10116	037016	012601				MOV	(SP)+,R1	;POP STACK INTO R1
10117	037020	012600				MOV	(SP)+,RO	;POP STACK INTO RO
10118	037022	000414				BR	6\$	;OSKI TO EXIT
10119	037024	032737	000200	001540	2\$:	BIT	#RDY,T.CS1	;TEST READY SET
10120	037032	001004				BNE	5\$	;YES - GOOD EXIT
10121	037034	012737	053732	001400		MOV	#EM7,EM3N	;MESSAGE (SUBSYS CLR NOT SET READY)
10122	037042	000753				BR	4\$	
10123	037044	012601			5\$:	MOV	(SP)+,R1	;RESTORE REGS
10124	037046	012600				MOV	(SP)+,RO	
10125	037050	062716	000002			ADD	#2,(SP)	;GOOD RETURN
10126	037054	000002			6\$:	RTI		
10127								
10128						.SBTTL	WORD COUNT AT END OF OPERATION CHECK	
10129						;	THIS ROUTINE COMPARES THE CONTENTS OF THE TEST STORAGE FOR	
10130						;	THE WORD COUNT AGAINST THE SUPPLIED VALUE. IF UNEQUAL, THE	
10131						;	ERROR FLAG (WCERR) IS SET IN GROUP 4 ERROR FLAGS (GRP4ER)	
10132						;		
10133						;		
10134						;		
10135						;		
10136	037056	012437	047424			CHKWC:	MOV	(R4)+,EXPWC
10137								;STORE EXPECTED VALUE
10138	037062	023737	047424	001542		CMP	EXPWC,T.WC	;COMPARE
10139	037070	001406				BEQ	1\$	;EQUAL - SKIP
10140	037072	052737	000001	047462		BIS	#WCERR,GRP4ER	;SET ERROR FLAG
10141	037100	013737	001542	047440		MOV	T.WC,REALWC	;STORE REAL WORD COUNT
10142	037106	000204			1\$:	RTS	R4	;RETURN.
10143								
10144						.SBTTL	BUS ADDRESS AT END OF OPERATION CHECK	
10145						;	THIS ROUTINE COMPUTES THE EXPECTED BUS ADDRESS AT THE END OF	
10146						;	A TRANSFER BY USING THE INITIAL BUS ADDRESS, ADDING IN THE	
10147						;	INITIAL WORD COUNT, AND SUBTRACTING ANY RESIDUAL WORD COUNT.	
10148						;	IF THIS COMPUTED BA DOES NOT EQUAL THE CONTENTS OF RKBA	
10149						;	AN ERROR FLAG (BAERR) IS SET IN GROUP 4 ERROR FIELD (GRP4ER)	
10150						;		
10151						;		
10152						;	IF BUS ADDRESS INCREMENT INHIBIT WAS SET, THE EXPECTED BUS	
10153						;	ADDRESS IS THE STARTING BUS ADDRESS.	
10154						;		
10155	037110	010046				CHKBA:	MOV	RO,-(SP)
10156	037112	010146					MOV	R1,-(SP)
10157	037114	010346					MOV	R3,-(SP)
10158	037116	032737	000020	001610		BIT	#BAI,L.CS2	;TEST IF BAI SET
10159	037124	001404				BEQ	4\$	;NO - SKIP
10160	037126	013737	001604	047430		MOV	L.BA,EXPBA	;STORE EXPECTED BA (SAME AS STARTING BA)
10161	037134	000441				BR	3\$	
10162	037136	013700	001602		4\$:	MOV	L.WC,RO	;GET INITIAL WORD COUNT
10163	037142	005400				NEG	RO	

H15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 188
BUS ADDRESS AT END OF OPERATION CHECK

SEQ 0188

10164	037144	113703	001601		MOVB	L.CS1+1,R3	;GET BA16 & BA17
10165	037150	042703	177774		BIC	#177774,R3	;CLEAR UNWANTED BITS
10166							
10167	037154	005700			TST	RO	;TEST IF INITIAL WORD COUNT 0
10168	037156	001003			BNE	6\$	;NO - SKIP
10169	037160	062703	000002		ADD	#2,R3	;ADD 2 TO BA16,17 (65K WORD XFER)
10170	037164	000477			BR	9\$	;SKIP
10171	037166	005700		6\$:	TST	RO	;TEST IF INITIAL WC BIT 15 SET
10172	037170	100001			BPL	5\$	;NO - SKIP
10173	037172	005203			INC	R3	;BUMP BA16,17 (32K WORD XFER)
10174	037174	006300		5\$:	ASL	RO	;SHIFT WORD COUNT TO MAKE MEM ADD CNT
10175	037176	063700	001604		ADD	L.BA,RO	;ADD IN START BUFFER ADD
10176	037202	005503			ADC	R3	;IF CARRY - ADD INTO BA16,17
10177	037204	013701	001542	9\$:	MOV	T.WC,R1	;GET END OF OPERATION WORD COUNT
10178	037210	001411			BEQ	1\$	;BRANCH IF ZERO
10179	037212	005401			NEG	R1	
10180	037214	005701			TST	R1	;TEST END OPERATION WC BIT 15 SET
10181	037216	100001			BPL	7\$	;NO - SKIP
10182	037220	005303			DEC	R3	;DEC BA 16,17 (32K WC LEFT)
10183	037222	006301		7\$:	ASL	R1	;SHIFT WC TO MAKE MEM ADD CNT
10184	037224	160100			SUB	R1,RO	;SUB FROM COMPUTED BUS ADDRESS
10185	037226	005603			SBC	R3	;SUB CARRY FROM BA16,17
10186	037230	010337	047426		MOV	R3,EXPUBA	;STORE EXPECTED UPPER BA BITS
10187	037234	010037	047430	1\$:	MOV	RO,EXPBA	
10188	037240	020037	001544	3\$:	CMP	RO,T.BA	;EQUAL TO COMPUTED?
10189	037244	001406			BEQ	2\$	;YES - SKIP
10190	037246	052737	000004	047462	BIS	#BAERR,GRP4ER	;ELSE SET BAERR FLAG
10191	037254	013737	001544	047444	MOV	T.BA,REALBA	;STORE REAL BUS ADDRESS
10192	037262	113703	001541		MOV	T.CS1+1,R3	;GET REAL UPPER BA
10193	037266	042703	177774	2\$:	BIC	#177774,R3	;CLEAR UNWANTED BITS
10194	037272	020337	047426		CMP	R3,EXPUBA	;CHECK IF EQUAL
10195	037276	001405			BEQ	8\$	;YES - SKIP
10196	037300	052737	000002	047462	BIS	#UBAERR,GRP4ER	;ELSE SET UBA ERROR
10197	037306	010337	047442		MOV	R3,REALUB	;STORE REAL UPPER BA
10198	037312	012603		8\$:	MOV	(SP)+,R3	
10199	037314	012601			MOV	(SP)+,R1	
10200	037316	012600			MOV	(SP)+,RO	
10201	037320	000204			RTS	R4	
10202							
10203					SBTTL	CYLINDER, TRACK, SECTOR TEST AT END OF OPERATION	
10204					*	THIS ROUTINE CHECKS THAT THE CONTENTS OF THE RKDCYL AND RKDA	
10205					*	ARE CORRECT FOR ANY SIZE DATA TRANSFER AT THE END OF THE	
10206					*	OPERATION. THE CONTENTS OF THE LOAD REGISTER STORAGE ARE	
10207					*	COUNTED ON TO HAVE THE INITIAL VALUES TO MAKE THE	
10208					*	NECESSARY CALCULATION.	
10209					*		
10210					*	ALL THREE VALUES ARE GENERATED AND STORED IN EXPECTED VALUES	
10211					*	STORAGE EXPCYL, EXPTRK, EXPSEC. ALL 3 ARE CHECKED AND	
10212					*	IF ONE OR MORE ARE WRONG, THE CORRESPONDING BIT IN THE	
10213					*	ERROR FLAGS FIELD (GRP4ER) IS SET.	
10214					*		
10215					*CALL:	JSR R4,CHKCTS	
10216							
10217	037322	104413			CHKCTS:	SAVREG	
10218	037324	013700	001602		MOV	L.WC,RO	;GET SPECIFIED WORD COUNT
10219	037330	005400			NEG	RO	;NEGATE IT

10220	037332	013701	001542		MOV	T.WC,R1	;GET END OF OPERATION WORD COUNT
10221	037336	001401			BEQ	10\$	;IF ZERO - SKIP
10222	037340	005401			NEG	R1	;NEGATE IT
10223	037342	160100		10\$:	SUB	R1,R0	;COMPUTE ACTUAL WORDS TRANSFERRED
10224	037344	005001			CLR	R1	;CLEAR R1 FOR COUNTING
10225				:	THE FOLLOWING CODE DETERMINES HOW MANY SECTORS OF DATA HAS BEEN		
10226				:	TRANSFERRED IN THE OPERATION. ONCE IT HAS COMPUTED THAT, THE		
10227				:	END OF OPERATION VALUES FOR THE CYLINDER, TRACK, AND SECTOR		
10228				:	IS CALCULATED.		
10229	037346	022700	000400	1\$:	CMP	#400,R0	
10230	037352	003004			BGT	2\$	
10231	037354	005201			INC	R1	
10232	037356	162700	000400		SUB	#400,R0	
10233	037362	000771			BR	1\$	
10234	037364	005700		2\$:	TST	R0	
10235	037366	001401			BEQ	3\$	
10236	037370	005201			INC	R1	
10237				:	AT THIS POINT R1 HAS A COUNT OF THE		
10238				:	NUMBER OF FULL SECTOR TRANSFER + 1 IF A		
10239				:	PARTIAL SECTOR WAS TRANSFERRED.		
10240	037372	012703	000026	3\$:	MOV	#26,R3	
10241	037376	032737	010000		BIT	#CFMT,L.CS1	
10242	037404	001402			BEQ	4\$	
10243	037406	012703	000024		MOV	#24,R3	
10244				:	R3 HAS BEEN SET UP WITH THE NUMBER		
10245				:	OF SECTORS IN A TRACK FOR THE FORMAT USED		
10246	037412	013737	001614	047432	4\$:	MOV	L.DCYL,EXPCYL
10247	037420	113704	001607		MOVB	L.DT,R4	;GET STARTING VALUES FOR CYLINDER
10248	037424	042704	177400		BIC	#177400,R4	;TRACK
10249	037430	113705	001606		MOVB	L.DS,R5	;SECTOR
10250	037434	042705	177400		BIC	#177400,R5	
10251	037440	005301			DEC	R1	;ADJUST COUNT FOR ZERO DETECT
10252	037442	005205		5\$:	INC	R5	;BUMP SECTOR COUNT
10253	037444	020503			CMP	R5,R3	;DID THIS MAKE SECTOR COUNT > 1 TRACK?
10254	037446	001010			BNE	6\$	;NO - SKIP
10255	037450	005005			CLR	R5	;ELSE CLEAR SECTOR COUNT
10256	037452	005204			INC	R4	;BUMP TRACK COUNT
10257	037454	022704	000003		CMP	#3,R4	;DID THIS MAKE TRK COUNT > 1 CYLINDER?
10258	037460	001003			BNE	6\$	;NO - SKIP
10259	037462	005004			CLR	R4	;ELSE CLEAR TRACK COUNT
10260	037464	005237	047432		INC	EXPCYL	;BUMP CYLINDER COUNT
10261	037470	005301		6\$:	DEC	R1	;DEC COUNT
10262	037472	001363			BNE	5\$	;IF ZERO - EXIT
10263	037474	010437	047436		MOV	R4,EXPTRK	;STORE EXPECTED TRACK
10264	037500	010537	047434		MOV	R5,EXPSEC	;STORE EXPECTED SECTOR (CYL ALREADY SLOW)
10265	037504	023737	001560	047432	CMP	T.DCYL,EXPCYL	;TEST IF CYLINDER OK
10266	037512	001403			BEQ	7\$	;YES - SKIP
10267	037514	052737	000010	047462	BIS	#CYLERR,GRP4ER	;NO - SET ERROR FLAG
10268	037522	120437	001547	7\$:	CMPB	R4,T.DA+1	;TEST TRACK OK
10269	037526	001403			BEQ	8\$	;YES - SKIP
10270	037530	052737	000020	047462	BIS	#TRKERR,GRP4ER	;NO - SET ERROR FLAG
10271	037536	120537	001546	8\$:	CMPB	R5,T.DA	;TEST SECTOR COUNT OK
10272	037542	001403			BEQ	9\$	;YES - SKIP
10273	037544	052737	000040	047462	BIS	#SECERR,GRP4ER	;USE SET ERROR FLAG
10274	037552	012700	047440	9\$:	MOV	#REALWC,R0	
10275	037556	013720	001542		MOV	T.WC,(R0)+	;STORE REAL WORD COUNT

10276 037562 013720 001544
10277 037566 013720 001560
10278 037572 113710 001547
10279 037576 005720
10280 037600 113710 001546
10281 037604 104414
10282 037606 000204
10283
10284
10285
10286
10287
10288
10289
10290
10291
10292
10293
10294
10295
10296
10297
10298
10299
10300
10301
10302
10303
10304
10305
10306
10307
10308
10309
10310
10311
10312
10313
10314
10315
10316
10317
10318
10319
10320
10321
10322
10323
10324
10325
10326
10327
10328
10329
10330
10331

MOV T.BA,(R0)+ ;STORE REAL BUS ADDRESS
MOV T.DCYL,(R0)+ ;STORE REAL CYLINDER ADDRESS
MOVB T.DA+1,(R0) ;STORE REAL TRACK ADDRESS
TST (R0)+
MOVB T.DA,(R0) ;STORE REAL SECTOR ADDRESS
RESREG
RTS R4

SBTTL 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.

CHKME: SAVREG
MOV (SP),R0 ;GET POINTER TO ERROR WORDS
MOV (R0)+,\$TMP10 ;STORE EXPECTED ERROR GROUP 1
MOV (R0)+,\$TMP11 ;STORE EXPECTED ERROR GROUP 2
MOV (R0)+,\$TMP12 ;STORE EXPECTED ERROR GROUP 3
MOV R0,(SP) ;STORE RETURN
MOV #-1,\$TMP13 ;SET FLAG - EXPECTED ERROR
BR CHKST

CHKOP: SAVREG
CLR \$TMP13 ;RESET EXPECTED ERROR FLAG

CHKST: TGETRK ;GET 611 REGS IO TRAP
CLR GRP1ER ;CLEAR ERROR FLAGS
CLR GRP2ER
CLR GRP3ER
CLR GRP4ER
CLR GPSUMF ;CLEAR SUMMARY FLAGS
BIT #CS1ERBIT,T.CS1 ;TEST IF ERROR SET IN CS1
BNE 4\$;YES - SKIP
BIT #CS2ERBIT,T.CS2 ;TEST IF ERROR SET IN CS2
BNE 4\$;YES - SKIP
BIT #DSERBIT,T.DS ;TEST IF ERROR SET IN DS
BNE 4\$;YES - SKIP
TST T.ER ;TEST IF ERROR SET IN ER
BNE 4\$;YES - SKIP
BIT #CERR,T.CS1 ;COMBINED ERROR SET?
BEQ 9\$;NO - SKIP
BIS #CERNER,GRP1ER ;SET ERROR FLAG IN GROUP 1
JMP 25\$;SKIP

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.

K15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 191
OPERATION CHECK ROUTINE

SEQ 0191

```

10332
10333
10334
10335
10336
10337
10338 037756 005737 001250
10339 037762 001402
10340 037764 000137 040502
10341 037770 013700 001540
10342 037774 042700 177741
10343 040000 022700 000020
10344 040004 002445
10345 040006 022700 000030
10346 040012 003042
10347 040014 004437 037056
10348 040020 000000
10349 040022 004437 037110
10350 040026 004437 037322
10351 040032 005737 047462
10352 040036 001430
10353 040040 016037 047600 060770
10354 040046 013700 047462
10355 040052 005001
10356 040054 006200
10357 040056 103402
10358 040060 005720
10359 040062 000774
10360 040064 016037 047464 001450
10361 040072 016037 047424 001202
10362 040100 016037 047440 001204
10363 040106 104414
10364 040110 012776 000010 000000
10365 040116 000002
10366 040120
10367 040120 104414
10368 040122 062716 000002
10369 040126 000002
10370
10371
10372
10373 040130 012700 047454
10374 040134 012701 001540
10375 040140 012703 001550
10376 040144 012704 001552
10377 040150 012705 001554
10378
10379 040154 051510
10380
10381 040156 042710 120701
10382
10383 040162 032711 020000
10384 040166 001402
10385 040170 052710 000001
10386
10387 040174 032714 000010

;
; 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.
95: TST $TMP13 ;TEST IF ERROR EXPECTED
    BEQ 625 ;NO - SKIP
    JMP 255 ;YES - JUMP
625: MOV T.CS1,R0 ;GET CS1
    BIC #177741,R0 ;CHECK IF OPERATION IS READ DATA,
    CMP #20,R0 ;WRITE DATA, OR WRITE CHECK. IF
    BLT 35 ;NOT SKIP ALL CHECKING IN GROUP
    CMP #30,R0 ;FOUR
    BGT 35
    JSR R4,CHKWC ;CHECK WORD COUNT
    ;WORD 0 ;EXPECTED WORD COUNT
    JSR R4,CHKBA ;CHECK BUS ADDRESS
    JSR R4,CHKCTS ;CHECK CYL, TRACK, & SECTOR
    TST GRP4ER ;ANY GROUP 4 ERRORS?
    BEQ 35 ;NO - SKIP
    MOV CMDLB(R0),DF010A ;LOAD ADDRESS OF COMMAND MESSAGE
    MOV GRP4ER,R0 ;PUT GROUP 4 ERROR FLAG IN R0
    CLR R1 ;CLEAR R1 FOR INDEX COUNTER
    ASR R0 ;SHIFT FLAGS - FIRST ONE ON RIGHT IS ERROR TO
    BCS 25 ;BE REPORTED, REST ARE IGNORED
    TST (R0)+ ;WHEN AN ERROR BIT IS FOUND
    BR 15 ;GET THE ERROR MESSAGE ASSOCIATED
    MOV GRP4MS(R0),EMION ;WITH IT AND SET ERROR TABLE ITEM TO
    MOV EXPWC(R0),SREG10 ;POINT TO THE MESSAGE. LOAD REG10 & 11
    MOV REALWC(R0),SREG11 ;WITH EXPECTED & IS VALUES
    RESREG ;RESTORE REGISTER
    MOV #10,2(SP) ;MAKE THE ERROR CALL POINT TO THE
    RTI ;RIGHT TABLE ENTRY & RETURN.
15:
25:
35: RESREG
    ADD #2,(SP) ;BUMP RETURN PAST ERROR
    RTI ;RETURN
;
; THE FOLLOWING CODE BUILDS THE GROUP 1,2, & 3 ERROR WORDS.
45: MOV #GRP1ER,R0 ;SET UP GENERAL REGISTER AS POINTER
    MOV #T.CS1,R1 ;CS1
    MOV #T.CS2,R3 ;CS2
    MOV #T.DS,R4 ;DS
    MOV #T.ER,R5 ;AND ER
    BIS (R5),(R0) ;SET ALL BITS IN GRP1ER THAT
    ;CORRESPOND TO ERROR BITS IN R5ER
    BIC #ILF!ECH!BSE!HVR!OPI!DCK,(R0) ;CLEAR ALL THAT DON'T BELONG GRP1
    BIT #SPAR,(R1) ;TEST IF SPAR SET
    BEQ 55 ;NO - SKIP
    BIS #SPARERR,(R0) ;SET SPAR ERROR FLAG
55: BIT #ACLO,(R4) ;TEST ACLO SET

```

L15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 192
OPERATION CHECK ROUTINE

SEQ 0192

10388	040200	001402		BEQ	6\$		;NO - SKIP
10389	040202	052710	000100	BIS	#ACLOERR, (R0)		;SET ACLO ERROR FLAG
10390							
10391	040206	032714	000020	6\$: BIT	#SPDLSS, (R4)		;TEST SPEED LOSS SET
10392	040212	001402		BEQ	7\$		;NO - SKIP
10393	040214	052710	000200	BIS	#SPDERR, (R0)		;SET SPEED LOSS ERROR FLAG
10394							
10395	040220	032714	000040	7\$: BIT	#DROT, (R4)		;TEST IF DROT SET
10396	040224	001402		BEQ	8\$		;NO - SKIP
10397	040226	052710	000400	BIS	#DROTERR, (R0)		;SET DROT ERROR FLAG
10398							
10399	040232	032711	100000	8\$: BIT	#CERR, (R1)		;TEST CERR ITSELF SET
10400	040236	001002		BNE	10\$		;YES - SKIP
10401	040240	032710	020000	BIT	#NCERWE, (R0)		;SET NO CERR WITH ERROR FLAG
10402							
10403	040244	012700	047456	10\$: MOV	#GRP2ER, R0		;SET POINTER TO GROUP 2 ERROR FLAGS
10404							
10405	040250	032715	000100	BIT	#ECH, (R5)		;TEST IF ECH SET
10406	040254	001402		BEQ	11\$		;NO - SKIP
10407	040256	052710	000001	BIS	#ECHERR, (R0)		;SET ECH FLAG
10408							
10409	040262	032715	100000	11\$: BIT	#DCK, (R5)		;TEST DCK SET
10410	040266	001402		BEQ	12\$		;NO - SKIP
10411	040270	052710	000002	BIS	#DCKERR, (R0)		;SET DCK ERROR FLAG.
10412							
10413	040274	032713	040000	12\$: BIT	#WCE, (R3)		;TEST WRITE CHECK ERROR
10414	040300	001402		BEQ	120\$		;NO - SKIP
10415	040302	052710	000004	BIS	#WCKERR, (R0)		;SET WCE BIT
10416	040306	032713	100000	120\$: BIT	#DLT, (R3)		;TEST DATA LATE
10417	040312	001402		BEQ	13\$		;NO - SKIP
10418	040314	052710	000010	BIS	#DLTERR, (R0)		;SET DLT ERROR FLAG
10419							
10420	040320	032715	020000	13\$: BIT	#OPI, (R5)		;TEST OPI SET
10421	040324	001402		BEQ	14\$		;NO - SKIP
10422	040326	052710	000020	BIS	#OPIERR, (R0)		;SET OPI ERROR FLAG
10423							
10424	040332	032715	000400	14\$: BIT	#HVRC, (R5)		;TEST HVRC SET
10425	040336	001402		BEQ	16\$		;NO - SKIP
10426	040340	052710	000040	BIS	#HVRCERR, (R0)		;SET HVRC FLAG
10427							
10428	040344	032715	000200	16\$: BIT	#BSE, (R5)		;TEST BSE ERROR FLAG
10429	040350	001402		BEQ	17\$		;NO - SKIP
10430	040352	052710	000100	BIS	#BSERR, (R0)		;SET BSE FLAG
10431							
10432	040356	012700	047460	17\$: MOV	#GRP3ER, R0		;SET POINTER TO GROUP 3 FLAGS
10433							
10434	040362	032713	010000	BIT	#NED, (R3)		;TEST NED SET
10435	040366	001402		BEQ	18\$		;NO - SKIP
10436	040370	052710	000001	BIS	#NEDERR, (R0)		;SET NED FLAG
10437							
10438	040374	032711	004000	18\$: BIT	#CTO, (R1)		;TEST CTO SET
10439	040400	001402		BEQ	19\$		;NO - SKIP
10440	040402	052710	000002	BIS	#CTOERR, (R0)		;SET CTO FLAG
10441							
10442	040406	032713	000400	19\$: BIT	#UFE, (R3)		;TEST UFE SET
10443	040412	001402		BEQ	20\$		;NO - SKIP

M15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 193
OPERATION CHECK ROUTINE

SEQ 0193

10444	040414	052710	000004		BIS	#UFERR,(R0)	;SET UFE FLAG
10445							
10446	040420	032713	001000	20\$:	BIT	#MDS,(R3)	;TEST MDS SET
10447	040424	001402			BEQ	21\$	;NO - SKIP
10448	040426	052710	000010		BIS	#MDSERR,(R0)	;SET MDS FLAG
10449							
10450	040432	032713	002000	21\$:	BIT	#PGE,(R3)	;TEST PGE SET
10451	040436	001402			BEQ	22\$	;NO - SKIP
10452	040440	052710	000020		BIS	#PGERR,(R0)	;SET PGE FLAG
10453							
10454	040444	032713	004000	22\$:	BIT	#NEM,(R3)	;TEST NEM SET
10455	040450	001402			BEQ	23\$	;NO - SKIP
10456	040452	052710	000040		BIS	#NEMERR,(R0)	;SET NEM FLAG
10457							
10458	040456	032713	020000	23\$:	BIT	#UPE,(R3)	;TEST UPE SET
10459	040462	001402			BEQ	24\$	;NO - SKIP
10460	040464	052710	000100		BIS	#UPERR,(R0)	;SET UPE FLAG
10461							
10462	040470	032715	000001	24\$:	BIT	#ILF,(R5)	;TEST ILF SET
10463	040474	001402			BEQ	25\$	;NO - SKIP
10464	040476	052710	000200		BIS	#ILFERR,(R0)	;SET ILF FLAG.
10465							
10466							
10467							
10468							
10469							
10470							
10471							
10472							
10473							
10474							
10475							
10476							
10477	040502	005737	001250	25\$:	TST	\$TMP13	;CHECK IF AN ERROR WAS EXPECTED
10478	040506	001423			BEQ	110\$	;NO - SKIP
10479	040510	012704	047454		MOV	#GRP1ER,R4	;GET ADDRESS OF ERROR
10480	040514	012705	001242		MOV	#TMP10,R5	;GET ADDRESS OF EXPECTED ERRORS
10481							
10482	040520	011500		26\$:	MOV	(R5),R0	;GET EXPECTED ERROR
10483	040522	011401			MOV	(R4),R1	;GET GROUP ERROR FLAGS
10484	040524	020001			CMP	R0,R1	;ARE THEY EQUAL?
10485	040526	001003			BNE	27\$	;NO - SKIP
10486	040530	005000			CLR	R0	;CLEAR EXPECTED ED
10487	040532	005001			CLR	R1	;CLEAR OCCURED
10488	040534	000403			BR	28\$	
10489							
10490	040536	010003		27\$:	MOV	R0,R3	;STORE EXPECTED ERRORS
10491	040540	040100			BIC	R1,R0	;RESET EXPECTED THAT OCCURRED
10492	040542	040301			BIC	R3,R1	;RESET OCCURRED THAT EXPECTED
10493	040544	010025		28\$:	MOV	R0,(R5)+	;STORE EXPECTED THAT DID NOT OCCUR
10494	040546	010124			MOV	R1,(R4)+	;STORE OCCURRED THAT WERE NOT EXPECTED
10495	040550	022705	001250		CMP	#TMP13,R5	;ALL GROUPS CHECKED.
10496	040554	001361			BNE	26\$	;NO - LOOP
10497							
10498							
10499							

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

THE FOLLOWING CODE:

N15

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 194
OPERATION CHECK ROUTINE

SEQ 0194

10500
10501
10502
10503
10504
10505
10506
10507
10508
10509
10510
10511
10512
10513
10514
10515
10516
10517
10518
10519
10520
10521
10522
10523
10524 040556 005004
10525 040560 005005
10526 040562 012700 001224
10527 040566 012701 001226
10528 040572 012703 047576
10529 040576 012710 060750
10530 040602 012711 001442
10531 040606 013746 047460
10532 040612 004437 035730
10533 040616 005716
10534 040620 001403
10535 040622 061605
10536 040624 052713 000004
10537
10538 040630 005726
10539 040632 005737 001250
10540 040636 001412
10541 040640 013746 001246
10542 040644 004437 035730
10543 040650 005716
10544 040652 001403
10545 040654 052713 000040
10546 040660 061604
10547
10548 040662 005726
10549 040664 013746 047456
10550 040670 004437 035730
10551 040674 005716
10552 040676 001407
10553 040700 052713 000002
10554 040704 061605
10555 040706 012710 060724

- 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 (DEPENDENT 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.

```

110$: CLR R4 ;CLEAR COUNTERS
      CLR R5
      MOV #STMP1,R0 ;LOAD POINTERS FOR TEMPORARY STORAGE OF ADDRESS
      MOV #STMP2,R1 ;WHERE ASCIZ ADDRESSES GO
      MOV #GPSUMF,R3 ;POINTERS TO GROUP SUMMARY FLAGS
      MOV #DF007A,(R0) ;PRESET FOR GRP3 ERR MESSAGE BUILD
      MOV #DH7N,(R1)
      MOV GRP3ER,-(SP) ;GET GROUP 3 ERRORS, PUT ON STACK
      JSR R4,BITCNT ;GO COUNT NUMBER AT ERRORS
      TST (SP) ;ANY ERRORS?
      BEQ 29$ ;NO - SKIP
      ADD (SP),R5 ;ADD IN ERROR TOTAL
      BIS #GRP3ST,(R3) ;SET BIT TO INDICATE GROUP 3 ERROR

29$: TST (SP)+ ;CLEAR OFF STACK
     TST STMP13 ;ERROR EXPECTED
     BEQ 31$ ;NO - SKIP
     MOV STMP12,-(SP) ;PUT GROUP 3 NOT RECEIVED ERRORS ON STACK
     JSR R4,BITCNT ;COUNT NUMBER OF ERRORS.
     TST (SP) ;WERE THERE ANY
     BEQ 30$ ;NO - SKIP
     BIS #GP3NR,(R3) ;SET GROUP 3 NOT RECEIVED ERROR FLAG
     ADD (SP),R4 ;ADD COUNT TO TOTAL THESE

30$: TST (SP)+ ;CLEAR OFF STACK
31$: MOV GRP2ER,-(SP) ;GET GROUP 2 ERRORS FOR COUNTING
     JSR R4,BITCNT ;COUNT BITS
     TST (SP) ;ANY SET?
     BEQ 32$ ;NO - SKIP
     BIS #GRP2ST,(R3) ;SET FLAG FOR GROUP 2 ERRORS
     ADD (SP),R5 ;ADD INTO TOTAL
     MOV #DF006A,(R0) ;STORE ADDRESS FOR BUILDING REPORT

```

B16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 195
OPERATION CHECK ROUTINE

SEQ 0195

10556	040712	012711	001432	MOV	#DH6N, (R1)	
10557						
10558	040716	005726		32\$: TST	(SP)+	; CLEAR OFF STACK
10559	040720	005737	001250	TST	\$TMP13	; ANY EXPECTED ERRORS
10560	040724	001416		BEQ	34\$	; NO - SKIP
10561	040726	013746	001244	MOV	\$TMP11, -(SP)	; GET GROUP 2 NOT RECEIVED ERRORS
10562	040732	004437	035730	JSR	R4, BITCNT	; COUNT NUMBER OF BITS
10563	040736	005716		TST	(SP)	; ANY SET?
10564	040740	001407		BEQ	33\$	; NO - SKIP
10565	040742	052713	000020	BIS	#GP2NR, (R3)	; SET FLAG FOR GROUP 2 NOT RECEIVED
10566	040746	061604		ADD	(SP), R4	; ADD INTO TOTAL
10567	040750	012710	060724	MOV	#DF006A, (R0)	; STORE ADDRESS FOR BUILDING REPORT
10568	040754	012711	001432	MOV	#DH6N, (R1)	
10569						
10570	040760	005726		33\$: TST	(SP)+	; CLEAR OFF STACK
10571	040762	013746	047454	34\$: MOV	GRP1ER, -(SP)	; GET GROUP 1 ERROR FLAGS
10572	040766	004437	035730	JSR	R4, BITCNT	; COUNT THE NUMBER OF BITS
10573	040772	005716		TST	(SP)	; ANY SET?
10574	040774	001407		BEQ	35\$	; NO - SKIP
10575	040776	052713	000001	BIS	#GRP1ST, (R3)	; SET FLAG FOR GROUP 1 ERRORS SET
10576	041002	061605		ADD	(SP), R5	; ADD INTO TOTAL
10577	041004	012710	060640	MOV	#DF004A, (R0)	; LOAD ADDRESS FOR BUILDING REPORT
10578	041010	012711	001412	MOV	#DH4N, (R1)	
10579	041014	005726		35\$: TST	(SP)+	; CLEAR OFF STACK
10580	041016	005737	001250	TST	\$TMP13	; ANY EXPECTED ERRORS?
10581	041022	001416		BEQ	60\$	; NO - SKIP
10582	041024	013746	001242	MOV	\$TMP10, -(SP)	; GET GROUP 1 NO RECEIVED ERROR
10583	041030	004437	035730	JSR	R4, BITCNT	; COUNT # OF BITS
10584	041034	005716		TST	(SP)	; ANY SET?
10585	041036	001407		BEQ	36\$	; NO - SKIP
10586	041040	052713	000010	BIS	#GP1NR, (R3)	; SET FLAG FOR GROUP 1 NOT RECEIVED
10587	041044	061604		ADD	(SP), R4	; ADD INTO TOTAL
10588	041046	012710	060640	MOV	#DF004A, (R0)	; LOAD ADDRESS FOR BUILDING REPORT
10589	041052	012711	001412	MOV	#DH4N, (R1)	
10590	041056	005726		36\$: TST	(SP)+	; CLEAR OFF STACK
10591	041060	032713	000011	60\$: BIT	#GRP1ST!GP1NR, (R3)	; ANY GROUP 1 ERROR
10592	041064	001414		BEQ	52\$	; NO - SKIP
10593	041066	042713	040000	BIC	#DRSTER, (R3)	
10594	041072	004437	036442	JSR	R4, GETDRS	
10595	041076	000401		BR	51\$	; ERROR RETURN
10596	041100	000406		BR	52\$	; NO ERROR RETURN
10597	041102	012710	060670	51\$: MOV	#DF005A, (R0)	; CHANGE TO FORMAT 5 - STORE ADDRESS
10598	041106	012711	001422	MOV	#DH5N, (R1)	; FOR BUILDING REPORT
10599	041112	052713	040000	BIS	#DRSTER, (R3)	; SET DRIVE STATUS ERROR
10600	041116			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
OF TABLE.

AT THIS TIME
R5 CONTAINS EITHER THE NUMBER OF ERRORS THAT OCCURRED BUT WERE
NOT EXPECTED OR

10601
10602
10603
10604
10605
10606
10607
10608
10609
10610
10611

C16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 196
OPERATION CHECK ROUTINE

SEQ 0196

```

10612      :
10613      :
10614      :
10615      :
10616      :
10617      :
10618      :
10619      :
10620      :
10621      :
10622      :
10623      :
10624      :
10625      :
10626      :
10627      :
10628      041116 032777 020000 14001
10629      041124 001402
10630      041126 000137 041514
10631      041132 005737 001250      37$:
10632      041136 001004
10633      :
10634      :
10635      :
10636      041140 012771 057643 000000
10637      041146 000411
10638      041150 012771 057464 000000      38$:
10639      041156 032713 000070
10640      041162 001003
10641      041164 012771 057561 000000
10642      041172 013701 001540      39$:
10643      041176 042701 177741
10644      :
10645      041202 016170 047600 000000
10646      :
10647      041210 032713 000007
10648      041214 001462
10649      :
10650      :
10651      :
10652      041216 013701 047460
10653      041222 012700 047500
10654      041226 005037 001252      140$:
10655      041232 012737 000021 001254      40$:
10656      041240 000241
10657      041242 006001      41$:
10658      041244 103406
10659      041246 062700 000002      141$:
10660      041252 005337 001254
10661      041256 001371
10662      041260 000411
10663      041262 011037 041274      42$:
10664      041266 104401 001273
10665      041272 104401
10666      041274 000000      43$:
10667      041276 005305

```

THE NUMBER OF ERRORS THE OCCURRED IF NONE WERE EXPECTED
R4 CONTAINS THE NUMBER OF ERRORS THAT WERE EXPECTED BUT
DID NOT OCCUR.
STMP10 CONTAINS GROUP 1 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
STMP11 CONTAINS GROUP 2 ERRORS THAT WERE EXPECTED BUT DID NOT OCCUR
STMP12 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)=#STMP2 CONTAINS THE ADDRESS OF THE HEADER MESSAGE ADDRESS IN
OF THAT MUST BE ALTERED TO IDENTIFY THE OPERATION
(R0)=#STMP1 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 #SW13,2SWR ;IS REPORT INHIBITED?
BEQ 37\$;NO - SKIP
JMP 49\$;ELSE EXIT
TST STMP13 ;WERE ERRORS EXPECTED?
BNE 38\$;YES - SKIP
IF NO ERRORS WERE EXPECTED, STMP10,11, &12 ARE NOT MEANINGFUL
MOV #DH007,2(R1) ;HEADER = ERROR IN OPERATION
BR 39\$
MOV #DH005,2(R1) ;PRESET NORMSG = 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
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
MOV #GRP3MS,R0 ;SET POINTER TO GRP3 ERROR MESSAGES
CLR STMP14 ;CLEAR GROUP PRINTING INDICATOR
MOV #17,STMP15 ;PRESET SHIFT COUNT
CLC ;CLEAR CARRY
ROR R1 ;ROTATE ERROR FLAGS
BCS 42\$;WAS BIT SHIFTED OUT SET?
ADD #2,R0 ;NO - BUMP POINTER
DEC STMP15 ;DEC SHIFT COUNT
BNE 41\$;LOOP IF SHIFT NOT ZERO
BR 44\$
MOV (R0),43\$;GET ERROR MESSAGE ADDRESS FROM TABLE
TYPE ,\$CRLF ;TYPE CRLF
TYPE ;TYPE ERROR MESSAGE
;MESSAGE ADDRESS GOES HERE
DEC R5 ;DECEREMENT TOTAL ERROR COUNT.

10668	041300	001362			BNE	1415		; LOOP IF ZERO
10669	041302	000427			BR	465		; ELSE EXIT GPR ERROR PRINT LOOP
10670								
10671	041304	005713		445:	TST	(R3)		; TEST GPSUMF FLAG FOR PRINTING ERROR NOT RECEIVED
10672	041306	100455			BMI	475		; YES - SKIP
10673	041310	005737	001252		TST	\$TMP14		; PRINTING GROUP 3?
10674	041314	001007			BNE	455		; NO - SKIP
10675	041316	013701	047456		MOV	GRP2ER,R1		; ELSE SET TO GROUP 2, GET GRP2ER
10676	041322	012700	047520		MOV	#GRP2MS,R0		; & SET POINTER TO GROUP 2 ERROR MESSAGE TABLE
10677	041326	005237	001252		INC	\$TMP14		; BUMP TO INDICATE PRINTING GROUP 2
10678	041332	000737			BR	405		; GO RESTART PRINT LOOP
10679	041334	022737	000002 001252	455:	CMP	#2,\$TMP14		; PRINTING GROUP 1?
10680	041342	001407			BEQ	465		; YES - EXIT GPR ERROR PRINT LOOP.
10681	041344	013701	047454		MOV	GRP1ER,R1		; ELSE SET TO GROUP 1, GET GROUP 1 ERROR
10682	041350	012700	047536		MOV	#GRP1MS,R0		; SET POINTER TO GROUP 1 ERROR MESSAGE TABLE
10683	041354	005237	001252		INC	\$TMP14		; BUMP TO INDICATE PRINTING GROUP 1
10684	041360	000724			BR	405		; RESTART PRINT LOOP.
10685								
10686	041362	005737	001250	465:	TST	\$TMP13		; EXPECTING ERRORS?
10687	041366	001452			BEQ	495		; NO - SKIP
10688								
10689								
10690								
10691	041370	005713			TST	(R3)		; TEST IF PRINTING NOT RECEIVED ERRORS
10692	041372	100423			BMI	475		; YES - SKIP
10693	041374	032713	000070		BIT	#GP1NR!GP2NR!GP3NR,(R3)		; ANY NOT RECEIVED ERRORS
10694	041400	001445			BEQ	495		; NO - SKIP
10695	041402	032713	000007		BIT	#GRP1ST!GRP2ST!GRP3ST,(R3)		; ANY NOT RECEIVED ERRORS?
10696	041406	001404			BEQ	1465		; NO - SKIP LABEL FOR UNEXPECTED ERRORS
10697	041410	104401	001273		TYPE	,SCLF		; TYPE CRLF
10698	041414	104401	057561		TYPE	,DH006		; TYPE HEADER FOR PREVIOUS ERRORS
10699	041420	052737	100000 047576	1465:	BIS	#REPNR,GPSUMF		; SET PRINTING NOT RECEIVED ERRORS SWITCH
10700	041426	010405			MOV	R4,R5		; MOVE TOTAL ERRORS TO R5
10701	041430	013701	001246		MOV	\$TMP12,R1		; GET GRP3 NOT RECEIVED ERRORS
10702	041434	012700	047500		MOV	#GRP3MS,R0		; SET POINTER TO GROUP 3 MESSAGES
10703	041440	000672			BR	1405		; GO START PRINT LOOP
10704	041442	005737	001252	475:	TST	\$TMP14		; PRINTING GROUP 3?
10705	041446	001007			BNE	485		; NO - SKIP
10706	041450	013701	001244		MOV	\$TMP11,R1		; ELSE SETUP TO PRINT GROUP 2 - GET ERRORS
10707	041454	012700	047520		MOV	#GRP2MS,R0		; & SET POINTER TO GROUP 2 MESSAGE TABLE
10708	041460	005237	001252		INC	\$TMP14		; BUMP TO INDICATE GROUP 2 PRINTING
10709	041464	000662			BR	405		; GO START PRINT LOOP
10710	041466	022737	000002 001252	485:	CMP	#2,\$TMP14		; PRINTING GROUP 1?
10711	041474	001407			BEQ	495		; YES - EXIT LOOP
10712	041476	013701	001242		MOV	\$TMP10,R1		; SET POINTER TO GROUP 1 MESSAGE
10713	041502	012700	047536		MOV	#GRP1MS,R0		; TABLE AND GET GROUP 1 ERRORS.
10714	041506	005237	001252		INC	\$TMP14		; BUMP TO INDICATE GROUP 1 PRINTING
10715	041512	000647			BR	405		; START LOOP AGAIN.
10716								
10717	041514	032713	000077	495:	BIT	#77,(R3)		; TEST IF ANY ERRORS TO BE REPORTED
10718								
10719								
10720	041520	001004			BNE	615		; GRP1ST!GRP2ST!GRP3ST
10721	041522	104414			RESREG			; GP1NR!GP2NR!GP3NR
10722	041524	062716	000002		ADD	#2,(SP)		; YES - SKIP
10723	041530	000002			RTI			; ELSE EXIT
								; BUMP FOR GOOD RETURN

E16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 198
OPERATION CHECK ROUTINE

SEQ 0198

```

10724
10725 041532 112776 000007 000000 61$: MOV B #7,2(SP) ;PRESET FOR GROUP 3 ERROR RETURN.
10726 041540 032713 000022          BIT #GRP2ST!GP2NR,(R3) ;ANY GROUP 2 ERRORS?
10727 041544 001403          BEQ 50$ ;NO - SKIP
10728 041546 112776 000006 000000          MOV B #6,2(SP) ;ELSE SET FOR GROUP 2 ERROR RETURN
10729
10730 041554 032713 000011          50$: BIT #GRP1ST!GP1NR,(R3) ;ANY GROUP 1 ERRORS?
10731 041560 001411          BEQ 53$ ;NO - SKIP
10732 041562 112776 000004 000000          MOV B #4,2(SP) ;ELSE SET FOR GROUP 1 ERROR RETURN.
10733 041570 032713 040000          BIT #DASTER,(R3) ;CHECK IF ERROR GETTING DRIVE STATUS
10734 041574 001403          BEQ 53$ ;NO - SKIP
10735 041576 112776 000005 000000          MOV B #5,2(SP) ;ELSE CHANGE RETURN FORM GROUP 1
10736
10737 041604 005737 001264          53$: TST $ESCAPE ;CHECK IF ESCAPE ALREADY SET
10738 041610 001011          BNE 54$ ;YES - SKIP
10739 041612 013700 001302          MOV $TESTN,R0 ;SET UP $ESCAPE TO FORCE
10740 041616 006300          ASL R0 ;ABORT TO PRESENT TEST AFTER
10741 041620 016037 033032 001264          MOV $SW08TB(R0), $ESCAPE ;ERROR IS REPORTED
10742 041626 162737 000002 001264          SUB #2, $ESCAPE ;BUT GO TO NEXT SCOPE STATEMENT
10743 041634 104414          54$: RESREG
10744 041636 000002          RTI ;RETURN
10745
10746 ;*****
10747 ;SBTTL BAD SECTOR CHECK
10748 ; THE FIELD WHOSE ADDRESS IS IN THE LOCATION AFTER THE
10749 ; CALL IS CHECKED TO SEE IF ANY SECTORS ARE LISTED THEREIN
10750 ; THAT HAVE THE CYLINDER AND TRACK ADDRESS SPECIFIED IN
10751 ; L.DCYL AND L.DT. IF A SECTOR IS FOUND IN THIS FIELD
10752 ; THAT IS BAD FOR THAT CYLINDER AND TRACK, THE SECTOR NUMBER
10753 ; IS PLACED ON THE STACK. THE TOTAL NUMBER OF BAD SECTORS
10754 ; IS PLACED ON THE STACK AFTER THE ENTIRE
10755 ; FIELD IS SEARCHED.
10756 ;
10757 ; CALL: JSR R4,BDSRCK
10758 ; <ADDRESS OF FIELD TO BE SEARCHED>
10759 ;*****
10760
10761 041640 012637 001236          BDSRCK: MOV (SP)+,$TMP6 ;STORE OLD R4 CONTENTS
10762 041644 010437 001240          MOV R4,$TMP7 ;GET RETURN ADDRESS
10763 041650 011404          MOV (R4),R4 ;GET POINTER TO FIELD TO BE CHECKED
10764 041652 005037 001234          CLR $TMP5 ;CLEAR A COUNTER
10765 041656 005714          1$: TST (R4) ;TEST IF FIELD HAS NO (OR NO MORE)ENTRIES
10766 041660 100417          BMI 4$ ;YES - EXIT
10767 041662 023724 001614          CMP L.DCYL,(R4)+ ;IS THIS ENTRY FOR THIS CYLINDER?
10768 041666 001012          BNE 3$ ;NO - SKIP
10769 041670 005204          INC R4 ;BUMP TO TRACK
10770 041672 123714 001607          CMPB L.DT,(R4) ;IS ENTRY FOR THIS TRACK?
10771 041676 001005          BNE 2$ ;NO - SKIP
10772 041700 005046          CLR -(SP) ;CLEAR STACK LOCATION
10773 041702 114416          MOVB -(R4), (SP) ;PUT SECTOR NUMBER ON STACK
10774 041704 005237 001234          INC $TMP5 ;BUMP COUNTER
10775 041710 000401          BR 3$ ;BRANCH
10776
10777 041712 005304          2$: DEC R4 ;DECREMENT POINTER TO WORD ALIGN
10778 041714 005724          3$: TST (R4)+ ;BUMP TO NEXT ENTRY
10779 041716 000757          BR 1$ ;TEST NEXT ENTRY

```

F16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 199
BAD SECTOR CHECK

SEQ 0199

10780
10781 041720 013746 001234
10782 041724 013746 001236
10783 041730 013704 001240
10784 041734 005724
10785 041736 000204
10786
10787
10788
10789
10790
10791
10792
10793
10794
10795
10796
10797
10798
10799
10800
10801
10802
10803
10804
10805
10806
10807
10808
10809
10810
10811
10812
10813
10814
10815
10816
10817
10818
10819
10820
10821
10822
10823
10824
10825
10826
10827
10828
10829
10830
10831
10832
10833
10834
10835

```

45:  MOV  STMP5, -(SP)      ;PUT COUNT ON STACK
      MOV  STMP6, -(SP)      ;PUT OLD R4 CONTENTS BACK ON STACK
      MOV  STMP7, R4         ;SET UP RETURN
      TST  (R4)+             ;BUMP PAST PARAMETER
      RTS   R4               ;RETURN

```

```

;*****
;SBTTL DATA GENERATION AND COMPARE ROUTINE
;
;CALLS:  JSR   R4, GENCOM
;        CONTROL WORD
;
;        JSR   R4, GENCOM
;        CONTROL WORD
;        LENGTH
;
;        JSR   R4, GENCOM
;        CONTROL WORD
;        RELOCATION CONSTANT
;        LENGTH
;
;RETURN:  RTS   R4
;
;R4 IS ADJUSTED IN THE CODE FOR THE FOLLOWING RETURNS:
;THE FIRST CALL RETURNS TO THE LOCATION FOLLOWING THE
;CONTROL WORD. THIS IS UNCONDITIONAL.
;
;THE SECOND CALL RETURNS TO THE LOCATION FOLLOWING THE LENGTH IF
;THE OPERATION REQUIRES DATA COMPARE AND DATA MISCOMPARED.
;IF DATA IS TO BE GENERATED ONLY OR NO DATA COMPARE
;ERRORS OCCURRED, THE RETURN IS TO LENGTH +4.
;
;THE THIRD CALL IS IDENTICAL TO THE SECOND.
;
;DEFINITION OF CONTROL WORD:
;BIT 15 - DO COMPARE OPERATION OF Ibuff (SOURCE) TO Obuff
;         (DESTINATION). EXPECTED VALUES ARE IN Obuff (DESTINATION).
;BIT 14 - RESUME COMPARE OPERATION FROM POINT LEFT BY LAST COMPARE.
;BIT 13 - INVOKE MEMORY MANAGEMENT FOR SOURCE (IBUFF).
;BIT 12 - INVOKE MEMORY MANAGEMENT FOR DESTINATION (OBUFF).
;BIT 11 - REPEAT FIRST WORD OF SELECTED PATTERN THROUGHOUT OBUFF.
;BIT 10 - CLEAR Ibuff TO PATTERN SELECTED.
;BIT 9 - BUILD HEADERS, CONSIDERING BS FILES
;BIT 8 - BUILD HEADERS, ALL SECTORS INDICATE GOOD SECTORS.
;BIT 7 - HEADER OPERATION SPECIFIED (EITHER COMPARE OR BUILD).
;BIT 6 TO 0 - PATTERN SELECT FIELD, OCTAL ENCODED. 0 INDICATES
;            NO DATA GENERATION, 1 IS ALL ZEROS, AND 7 IS ALL ONES.
;            OTHER PATTERNS PROVIDED ARE PATTERNS 2-6, 8-16.
;
;EXPLANATION OF CALLS:
;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
MOV    #EM54,EM15N    ;; PRESET FOR HEADER COMPARE ERROR
BIT    #BIT7,R0       ;; HEADER OPERATION SPECIFIED?
BNE    18$           ;; YES - SKIP
MOV    #EM55,EM15N    ;; CHANGE FOR DATA COMPARE ERROR
JMP    17$           ;; ELSE JUMP TO DATA ROUTINE

18$:   MOV    R4,-(SP)      ;; PUSH R4 ON STACK
      BIT    #BIT8:BIT9,R0 ;; MUST HEADERS BE BUILT?
      BNE    19$           ;; YES - SKIP
      JMP    11$           ;; ELSE JUMP TO HEADER COMPARE
19$:   MOVB   L.DT,R1      ;; START HEADER BUILD ROUTINE
      MOV    L.DCYL,R3    ;; GET TRACK AND CYL
      MOV    #5,R5        ;; SET COUNT TO SHIFT TRACK FOR HDR WORD

```

```

10836
10837
10838
10839
10840
10841
10842
10843
10844
10845
10846
10847
10848
10849
10850
10851
10852
10853
10854
10855
10856
10857
10858
10859
10860
10861
10862
10863
10864
10865
10866
10867
10868
10869
10870
10871
10872
10873 041740
10874 041740 010046
10875 041742 010146
10876 041744 010346
10877 041746 010546
10878 041750 012400
10879 041752 012737 056210 001520
10880 041760 032700 000200
10881 041764 001005
10882 041766 012737 056237 001520
10883 041774 000137 042526
10884 042000
10885 042000 010446
10886 042002 032700 001400
10887 042006 001002
10888 042010 000137 042310
10889 042014 113701 001607
10890 042020 013703 001614
10891 042024 012705 000005

```

H16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 201
DATA GENERATION AND COMPARE ROUTINE

SEQ 0201

10892											
10893	042030	006301			1\$:	ASL	R1				;SHIFT TRACK
10894	042032	005305				DEC	R5				;DECREMENT TRACK
10895	042034	001375				BNE	1\$				;LOOP UNTIL COUNT 0
10896											
10897	042036	012704	000026			MOV	#26,R4				;PRESET FOR 26 SECTOR MODE
10898	042042	032737	010000	001600		BIT	#CFMT,L.CS1				;IS IT 24 SECTOR MODE?
10899	042050	001404				BEQ	2\$				;NO - SKIP
10900	042052	012704	000024			MOV	#24,R4				;CHANGE COUNT FOR 24 SECTOR MODE
10901	042056	052701	001000			BIS	#BIT9,R1				;SET 24 SECTOR MODE BIT IN WRD 2 OF HDR
10902											
10903	042062	052701	140000		2\$:	BIS	#BIT15:BIT14,R1				;SET BS BITS TO INDICATE GOOD SECTOR
10904	042066	012705	063414			MOV	#OBUF,R5				;SET POINTER TO ADDRESS WHERE HEADERS GO
10905	042072	010325			3\$:	MOV	R3,(R5)+				;INSERT CYLINDER
10906	042074	010125				MOV	R1,(R5)+				;INSERT TRACK AND SECTOR
10907	042076	010337	001224			MOV	R3,STMP1				;CALCULATE HVRC WORD
10908	042102	010115				MOV	R1,(R5)				
10909	042104	040137	001224			BIC	R1,STMP1				
10910	042110	040315				BIC	R3,(R5)				
10911	042112	053725	001224			BIS	STMP1,(R5)+				;COMPLETE HVRC WORD INSERTION
10912											
10913	042116	005304				DEC	R4				;DECREMENT HEADER COUNT
10914	042120	001402				BEQ	4\$				;DONE? - YES, SKIP
10915	042122	005201				INC	R1				;BUMP SECTOR
10916	042124	000762				BR	3\$				;LOOP
10917											
10918	042126	032700	001000		4\$:	BIT	#BIT9,R0				;MUST HEADERS BE CORRECTED FOR TABLE ENTRIES?
10919	042132	001003				BNE	5\$				;YES - SKIP
10920	042134	005700			10\$:	TST	R0				;IS THIS A COMPARE OPERATION?
10921	042136	100464				BMI	11\$				;YES-GO DO HDR COMPARE
10922	042140	000534				BR	50\$				;ELSE GET OUT
10923											
10924	042142	013737	001640	042200	5\$:	MOV	BSF26P,6\$				;PRESET FOR BS FACTORY LIST
10925	042150	012737	100000	001224		MOV	#BIT15,STMP1				;SET BIT TO BE RESET IN BAD HEADER
10926	042156	032737	010000	001600		BIT	#CFMT,L.CS1				;IS THIS 26 SECTOR MODE?
10927	042164	001403				BEQ	8\$				;YES - SKIP
10928	042166	013737	001636	042200		MOV	BSF24P,6\$				;ELSE CHANGE FOR 24 SECTOR MODE
10929											
10930	042174	004437	041640		8\$:	JSR	R4,BDSRCK				;GO CHECK FOR BAD SECTOR THIS ADDRESS
10931	042200	000000			6\$:	.WORD	0				;POINTER TO FILE TO BE CHECKED GOES HERE
10932	042202	012605				MOV	(SP)+,R5				;GET # OF BAD SECTORS THIS PACK ADDRESS
10933	042204	001417				BEQ	9\$				;SKIP IF ZERO
10934											
10935	042206	011601			7\$:	MOV	(SP),R1				;GET 1ST BAD SECTOR NUMBER
10936	042210	006301				ASL	R1				;MULTIPLY SECTOR NUMBER BY 6 TO
10937	042212	006301				ASL	R1				;LOCATE SECTOR TO BE MARKED BAD
10938	042214	061601				ADD	(SP),R1				
10939	042216	062601				ADD	(SP)+,R1				
10940	042220	062701	000002			ADD	#2,R1				;ADD 2 FOR 2ND WORD THAT SECTOR
10941	042224	043761	001224	063414		BIC	STMP1,OBUF(R1)				;CLEAR BIT FOR BAD SECTOR IN HDR
10942	042232	043761	001224	063416		BIC	STMP1,OBUF+2(R1)				;CORRECT THE HVRC BIT
10943	042240	005305				DEC	R5				;DECREMENT BAD SECTOR COUNT
10944	042242	001361				BNE	7\$				;LOOP IF NOT ZERO
10945											
10946	042244	032737	100000	001224	9\$:	BIT	#BIT15,STMP1				;WERE WE DOING BS FACTORY LIST?
10947	042252	001730				BEQ	10\$				;NO - GO CHECK IF COMPARE MUST BE DONE

```

10948 042254 012737 040000 001224      MOV      #BIT14,$TMP1      ;ELSE SET BIT TO BE RESET IN BAD HDR
10949 042262 013737 001644 042200      MOV      BSS26P,6$      ;PRESET POINTER FOR 26 SECTOR MODE
10950 042270 032737 010000 001600      BIT      #CFMT,L.CS1    ;TEST IF WE ARE DOING 26 SECTOR MODE
10951 042276 001736                BEQ      8$              ;YES - SKIP TO START CHECK
10952 042300 013737 001642 042200      MOV      BSS24P,6$      ;CHANGE POINTER TOR 24 SECTOR MODE
10953 042306 000732                BR       8$              ;SKIP TO START CHECK
10954
10955                                ;START OF COMPARE
10956 042310 012701 000102                ;11$: MOV      #102,R1      ;PRESET FOR 102 WORDS OF HEADER
10957 042314 032737 010000 001600      BIT      #CFMT,L.CS1    ;CHECK IF 26 SECTOR MODE
10958 042322 001402                BEQ      12$            ;YES - SKIP
10959 042324 012701 000074                MOV      #74,R1      ;CHANGE TO 74 WORDS OF HEADER
10960
10961 042330 012704 061414                12$: MOV      #IBUFF,R4      ;SET START OF HEADERS TO BE COMPARED
10962 042334 012705 063414                MOV      #OBUFF,R5      ;SET START OF GOOD HEADERS
10963 042340 005003                CLR      R3              ;CLEAR COUNTER
10964 042342 032700 040000                BIT      #BIT14,R0      ;IS THIS A CONTINUATION OF EARLIER COMPARE
10965 042346 001412                BEQ      13$            ;NO - SKIP
10966 042350 013705 001700                28$: MOV      DESHLD,R5      ;GET VALUES WHERE PREVIOUS CHECK STOPPED
10967 042354 013704 001702                MOV      SRCHLD,R4      ;DESTINATION AND SOURCE
10968 042360 013703 001704                MOV      WRDNUM,R3      ;WORD NUMBER IN ERROR
10969 042364 013701 001706                MOV      WRDCNT,R1      ;WORD COUNT LEFT IN COMPARE
10970 042370 005701                TST      R1              ;TEST IF WORD COUNT LEFT = 0
10971 042372 001417                BEQ      50$            ;YES - EXIT
10972
10973 042374 032700 030000                13$: BIT      #BIT12!BIT13,R0 ;MEM MANAGE REQUIRED?
10974 042400 001402                BEQ      25$            ;NO - SKIP
10975 042402 005237 177572                INC      @#SRO          ;TURN IT ON
10976 042406 022425                25$: CMP      (R4)+,(R5)+ ;COMPARE THE WORDS
10977 042410 001012                BNE      14$            ;SKIP IF NOT EQUAL
10978 042412 005203                INC      R3              ;BUMP WORD NUMBER IN ERROR
10979 042414 005301                DEC      R1              ;DEC WORD COUNT LEFT IN COMPARE
10980 042416 001373                BNE      25$            ;LOOP IF NOT ZERO
10981 042420 032700 030000                BIT      #BIT12!BIT13,R0 ;MEM MANAGE IN USE?
10982 042424 001402                BEQ      50$            ;NO - SKIP
10983 042426 005337 177572                DEC      @#SRO          ;TURN IT OFF
10984
10985 042432 012604                50$: MOV      (SP)+,R4      ;;POP STACK INTO R4
10986 042434 000427                BR       16$
10987
10988                                ; ERROR REPORT PREP AND PARAMETER STORAGE FOR CONTINUATION
10989
10990 042436 010537 001700                14$: MOV      R5,DESHLD    ;STORE DESTINATION
10991 042442 010437 001702                MOV      R4,SRCHLD      ;SOURCE
10992 042446 014537 001202                MOV      -(R5),$REG10    ;LOAD GOOD WORD FOR REPORT
10993 042452 014437 001204                MOV      -(R4),$REG11    ;BAD WORD
10994 042456 010337 001206                MOV      R3,$REG12      ;WORD NUMBER
10995 042462 005301                DEC      R1              ;DEC COUNT LEFT FOR CONTINUATION
10996 042464 005203                INC      R3              ;BUMP BAD WORD NUMBER
10997 042466 010137 001706                MOV      R1,WRDCNT      ;STORE COUNT LEFT
10998 042472 010337 001704                MOV      R3,WRDNUM      ;WORD NUM IN ERROR
10999 042476 032700 030000                BIT      #BIT12!BIT13,R0 ;MEM MANAGE IS USE?
11000 042502 001402                BEQ      15$            ;NO - SKIP
11001 042504 005337 177572                DEC      @#SRO          ;TURN IT OFF
11002
11003 042510                15$:

```

J16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 203
DATA GENERATION AND COMPARE ROUTINE

SEQ 0203

```

11004 042510 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
11005 042512 005724      TST      (R4)+      ;;ERROR RETURN
11006
11007 042514      16$:      MOV      (SP)+,R5      ;;POP STACK INTO R5
11008 042514 012605      MOV      (SP)+,R3      ;;POP STACK INTO R3
11009 042516 012603      MOV      (SP)+,R1      ;;POP STACK INTO R1
11010 042520 012601      MOV      (SP)+,R0      ;;POP STACK INTO R0
11011 042522 012600      RTS      R4
11012 042524 000204
11013
11014 ;      DATA PATTERN PROCESSING ROUTINE
11015
11016 042526 032700 040000  17$:      BIT      #BIT14,R0      ;;CONTINUE WITH COMPARE?
11017 042532 001402      BEQ      29$      ;;NO - SKIP
11018 042534 010446      MOV      R4,-(SP)      ;;STORE RETURN
11019 042536 000704      BR      28$      ;;GO CONTINUE COMPARE
11020
11021 042540 012705 063414  29$:      MOV      #OBUFF,R5      ;;GET DESTINATION
11022 042544 012703 061414      MOV      #IBUFF,R3      ;;GET SOURCE
11023 042550 032700 030000      BIT      #BIT12!BIT13,R0  ;;USE MEM MANAGE?
11024 042554 001412      BEQ      21$      ;;NO - SKIP
11025
11026 042556 012437 172354      MOV      (R4)+,2#KIPAR6  ;;LOAD PAR FOR RELOCATION
11027 042562 032700 010000      BIT      #BIT12,R0      ;;RELOCATE SOURCE?
11028 042566 001403      BEQ      20$      ;;NO - SKIP
11029 042570 012705 140070      MOV      #140070,R5      ;;SET DESTINATION TO USE PAR6 + OFFSET
11030 042574 000402      BR      21$      ;;SKIP
11031 042576 012703 140070  20$:      MOV      #140070,R3      ;;SET SOURCE TO USE PAR6 + OFFSET
11032
11033 042602 012401  21$:      MOV      (R4)+,R1      ;;STORE COUNT
11034 042604 010446      MOV      R4,-(SP)      ;;STORE RETURN
11035 042606 010304      MOV      R3,R4      ;;PUT IN IBUFF POINTER
11036 042610 005003      CLR      R3      ;;CLEAR R3 FOR WORD NUMBER COUNTER
11037 042612 032700 000077      BIT      #77,R0      ;;ANY DATA PATTERN SPECIFIED?
11038 042616 001666      BEQ      13$      ;;NO - GO DO COMPARE
11039
11040 ;      START OF GENERATION ROUTINE
11041
11042 042620 010537 001700      MOV      R5,DESHLD      ;;STORE PARAMETERS FOR COMPARE
11043 042624 010437 001702      MOV      R4,SRCHLD
11044 042630 010337 001704      MOV      R3,WRDNUM
11045 042634 010137 001706      MOV      R1,WRDCNT
11046
11047 ;      CODE TO GENERATE DATA PATTERN IN AREA POINTED TO BY R5.
11048 ;      MEMORY MANAGEMENT WILL BE TURNED ON BUT RELOCATION
11049 ;      WILL NOT OCCUR UNLESS REQUIRED BY SWITCHES
11050
11051 042640 032700 030000      BIT      #BIT12!BIT13,R0  ;;MEMORY MANAGEMENT REQUIRED?
11052 042644 001402      BEQ      33$      ;;NO - SKIP
11053 042646 005237 177572      INC      2#SRO      ;;TURN IT ON
11054 042652 032700 002000  33$:      BIT      #BIT10,R0      ;;GENERATE PATTERN IN IBUFF?
11055 042656 001403      BEQ      32$      ;;NO - SKIP
11056 042660 010446      MOV      R4,-(SP)      ;;ELSE SWAP R4 AND R5
11057 042662 010504      MOV      R5,R4
11058 042664 012605      MOV      (SP)+,R5
11059

```

K16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 204
DATA GENERATION AND COMPARE ROUTINE

SEQ 0204

11060	042666	122700	000001	32\$:	CMPB	#1,R0	;PATTERN 1 (ALL ZEROS)?
11061	042672	001004			BNE	55\$	;NO - SKIP
11062	042674	005025		30\$:	CLR	(R5)+	;CLEAR WORD IN BUFF
11063	042676	005301			DEC	R1	;DEC WORD COUNT
11064	042700	001375			BNE	30\$	;LOOP UNTIL WORD COUNT ZERO
11065	042702	000550			BR	22\$	;EXIT BUILD
11066							
11067	042704	122700	000007	55\$:	CMPB	#7,R0	;PATTERN 7 (ALL ONES)?
11068	042710	001005			BNE	56\$	;NO - SKIP
11069	042712	012725	177777	31\$:	MOV	#-1,(R5)+	;LOAD WORD IN BUFF
11070	042716	005301			DEC	R1	;DEC WORD COUNT
11071	042720	001374			BNE	31\$	;LOOP UNTIL WORD COUNT ZERO
11072	042722	000540			BR	22\$	;EXIT BUILD
11073							
11074	042724	122700	000002	56\$:	CMPB	#2,R0	;PATTERN 2 SET UP
11075	042730	001003			BNE	57\$	
11076	042732	012703	046624		MOV	#PAT02,R3	
11077	042736	000504			BR	70\$	
11078							
11079	042740	122700	000003	57\$:	CMPB	#3,R0	;PATTERN 3 SET UP
11080	042744	001003			BNE	58\$	
11081	042746	012703	046664		MOV	#PAT03,R3	
11082	042752	000476			BR	70\$	
11083							
11084	042754	122700	000004	58\$:	CMPB	#4,R0	;PATTERN 4 SET UP
11085	042760	001003			BNE	59\$	
11086	042762	012703	046724		MOV	#PAT04,R3	
11087	042766	000470			BR	70\$	
11088							
11089	042770	122700	000005	59\$:	CMPB	#5,R0	;PATTERN 5 SET UP
11090	042774	001003			BNE	60\$	
11091	042776	012703	046764		MOV	#PAT05,R3	
11092	043002	000462			BR	70\$	
11093							
11094	043004	122700	000006	60\$:	CMPB	#6,R0	;PATTERN 6 SET UP
11095	043010	001003			BNE	61\$	
11096	043012	012703	047024		MOV	#PAT06,R3	
11097	043016	000454			BR	70\$	
11098							
11099	043020	122700	000010	61\$:	CMPB	#10,R0	;PATTERN 10 SET UP
11100	043024	001003			BNE	62\$	
11101	043026	012703	047064		MOV	#PAT10,R3	
11102	043032	000446			BR	70\$	
11103							
11104	043034	122700	000011	62\$:	CMPB	#11,R0	;PATTERN 11 SET UP
11105	043040	001003			BNE	63\$	
11106	043042	012703	047124		MOV	#PAT11,R3	
11107	043046	000440			BR	70\$	
11108							
11109	043050	122700	000012	63\$:	CMPB	#12,R0	;PATTERN 12 SET UP
11110	043054	001003			BNE	64\$	
11111	043056	012703	047164		MOV	#PAT12,R3	
11112	043062	000432			BR	70\$	
11113							
11114	043064	122700	000013	64\$:	CMPB	#13,R0	;PATTERN 13 SET UP
11115	043070	001003			BNE	65\$	

L16

RX611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 205
DATA GENERATION AND COMPARE ROUTINE

SEQ 0205

11116	043072	012703	047224		MOV	#PAT13,R3	
11117	043076	000424			BR	70\$	
11118							
11119	043100	122700	000014	65\$:	CMPB	#14,R0	;PATTERN 14 SET UP
11120	043104	001003			BNE	66\$	
11121	043106	012703	047264		MOV	#PAT14,R3	
11122	043112	000416			BR	70\$	
11123							
11124	043114	122700	000015	66\$:	CMPB	#15,R0	;PATTERN 15 SET UP
11125	043120	001003			BNE	67\$	
11126	043122	012703	047324		MOV	#PAT15,R3	
11127	043126	000410			BR	70\$	
11128							
11129	043130	122700	000016	67\$:	CMPB	#16,R0	;PATTERN 16 SET UP
11130	043134	001003			BNE	68\$	
11131	043136	012703	047364		MOV	#PAT16,R3	
11132	043142	000402			BR	70\$	
11133							
11134	043144	012703	047364	68\$:	MOV	#PAT16,R3	;SET UP FOR 16
11135							
11136	043150	032700	004000	70\$:	BIT	#BIT11,R0	;FIRST WORD REPEAT?
11137	043154	001020			BNE	73\$	;YES - SKIP
11138	043156	010446			MOV	R4,-(SP)	;STORE R4
11139	043160	010046			MOV	R0,-(SP)	;STORE R0
11140	043162	012700	000020		MOV	#16,R0	;PRESET COUNT FOR PATTERN LENGTH
11141	043166	010504			MOV	R5,R4	;STORE START OF BUFF
11142							
11143	043170	012325		71\$:	MOV	(R3)+,(R5)+	;MOV WORD TO BUFF
11144	043172	005301			DEC	R1	;DEC WORD COUNT
11145	043174	001405			BEQ	74\$	;EXIT IF ZERO
11146	043176	005300			DEC	R0	;DEC PAT LENGTH COUNT
11147	043200	001373			BNE	71\$	;LOOP IF NOT ZERO
11148							
11149	043202	012425		72\$:	MOV	(R4)+,(R5)+	;REPEAT PATTERN IN BUFFER
11150	043204	005301			DEC	R1	;DEC WORD COUNT
11151	043206	001375			BNE	72\$	;LOOP UNTIL WORD COUNT ZERO
11152							
11153	043210	012600		74\$:	MOV	(SP)+,R0	;RESTORE R0
11154	043212	012604			MOV	(SP)+,R4	;RESTORE R4
11155	043214	000403			BR	22\$	;EXIT BUILD
11156							
11157	043216	011325		73\$:	MOV	(R3),(R5)+	;MOV THE SAME WORD INTO BUFFER
11158	043220	005301			DEC	R1	;DEC WORD COUNT
11159	043222	001375			BNE	73\$	;LOOP UNTIL ZERO
11160							
11161	043224	032700	030000	22\$:	BIT	#BIT12!BIT13,R0	;MEMORY MANAGEMENT REQUIRED?
11162	043230	001402			BEQ	34\$	;NO - SKIP
11163	043232	005337	177572		DEC	#SRO	;TURN OFF MEM MANAGEMENT
11164	043236	005700		34\$:	TST	R0	;IS COMPARE REQUIRED?
11165	043240	100012			BPL	23\$	;NO - SKIP
11166	043242	013705	001700		MOV	DESHLD,R5	;RESTORE COMPARE PARAMETERS
11167	043246	013704	001702		MOV	SRCHLD,R4	
11168	043252	013703	001'04		MOV	WRDNUM,R3	
11169	043256	013701	001'06		MOV	WRDCNT,R1	
11170	043262	000137	042374		JMP	13\$	;GO START COMPARE
11171	043266			23\$:			

M16

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 206
DATA GENERATION AND COMPARE ROUTINE

SEQ 0206

```

11172 043266 012604      MOV      (SP)+,R4      ;:POP STACK INTO R4
11173 043270 000137 042514  JMP      16$      ;:GO TO EXIT
11174
11175 ;:*****
11176 ;:SBTTL PHASE LOCK LOOP CLOCK ADJUSTMENT ROUTINE
11177 ;:*
11178 ;:* THIS ROUTINE IS ENTERED VIA A START AT LOCATION 220(8). THE
11179 ;:* PROGRAM FIRST RUNS TEST 1, 2, AND 3 TO SET UP THE INTERNAL
11180 ;:* PROGRAM VARIABLES AND THEN JUMPS TO THE CLOCK ADJUST ROUTINE.
11181 ;:* THE ROUTINE SELECTS THE FIRST AVAILABLE DRIVE AND SETS AND
11182 ;:* RESETS DIAGNOSTIC MODE BIT IN MRI. INSTRUCTIONS ON WHERE TO
11183 ;:* SCOPE AND WHAT TO ADJUST ARE TYPED ON THE CONSOLE.
11184 ;:*
11185 ;:* THIS ROUTINE WILL LOOP UNTIL THE PROCESSOR IS HALTED.
11186 043274 104401 052650  ADJCLK: TYPE      ,OPR019      ;:TYPE ADJUSTMENT INSTRUCTIONS
11187
11188 043300 012762 000040 000010  MOV      #SCLR,RKCS2(R2) ;:CLEAR SUBSYSTEM
11189 043306 013762 001626 000010  MOV      DRVNUM,RKCS2(R2) ;:SELECT DRIVE
11190 043314 012762 000001 000000  MOV      #1,RKCS1(R2)
11191 043322 032762 000200 000000  5$: BIT      #RDY,RKCS1(R2) ;:WAIT FOR READY
11192 043330 001774          BEQ      5$
11193 043332 032737 100000 001664  BIT      #LCLKPR,OPTFLG ;:TEST IF CLOCK PRESENT
11194 043340 001402          BEQ      1$ ;:NO - SKIP
11195 043342 005077 136342          CLR      #KWLADD ;:CLEAR INTERRUPT ENABLE
11196
11197 043346 012762 000040 000026  1$: MOV      #DMD,RKMR1(R2) ;:SET DIAG MODE
11198 043354 012701 000014          MOV      #12.,R1 ;:SET A COUNT
11199 043360 005301          2$: DEC      R1 ;:DEC COUNT
11200 043362 001376          BNE      2$ ;:LOOP UNTIL ZERO
11201 043364 012762 000000 000026  MOV      #0,RKMR1(R2) ;:CLEAR MRI
11202 043372 012701 000014          MOV      #12.,R1 ;:SET COUNT
11203 043376 005301          3$: DEC      R1 ;:DEC COUNT
11204 043400 001376          BNE      3$ ;:LOOP UNTIL ZERO
11205 043402 000761          BR       1$ ;:RESTART LOOP
11206
11207 ;:SBTTL CONTROLLED HALT SUBROUTINE
11208 ;:*
11209 ;:* THIS ROUTINE IS ENTERED WHEN A CONTROL C IS TYPED. THE
11210 ;:* SUBSYSTEM IS CLEARED, THE DRIVE IS RECALIBRATED, AND, IF
11211 ;:* NECESSARY, CERTAIN CYLINDERS ARE REFORMATTED. THE REFORMATTING
11212 ;:* IS CONTROLLED BY THE LOCATION REFM1 WHICH CONTAINS THE ADDRESS
11213 ;:* OF THE CYLINDER TO BE REFORMATTED.
11214 043404 012737 000112 001302  CTRHLT: MOV      #STN,$TESTN ;:SET UP FOR HALT FAIL
11215
11216 043412 104416          TSS:VIT ;:CLEAR SUBSYSTEM
11217 043414 104003          ERROR 3 ;:BAD INIT ERROR
11218
11219 043416 113700 001102          MOV      $STNM,R0 ;:GET CURRENT TEST NUMBER
11220 043422 042700 177400          BIC      #177400,R0 ;:CLEAR UNUSED BITS
11221 043426 022700 000003          CMP      #3,R0 ;:TEST IF TEST NUMBER 3
11222 043432 001464          BEQ      PROGEND ;:GO TO HALT PROG
11223 043434 004437 035530          JSR      R4,LRLoad ;:LOAD "L" REGS
11224 043440 000113          RECAL ;:RECAL
11225 043442 000000          0 ;:0 WORDS
11226 043444 000000          0 ;:0 IS BUFF ADDRESS
11227 043446          .BYTE 0 ;:SECTOR 0

```

B01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 207
CONTROLLED HALT SUBROUTINE

SEQ 0207

11228	043447	000		.BYTE	0		;TRACK 0
11229	043450	000000			0		;CYLINDER 0
11230							
11231	043452	104417		TLOADRK			;LOAD RK REGS
11232	043454	104423		TWAT16			;WAIT FOR INTERRUPT
11233	043456	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
11234							
11235	043460	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11236	043462	10400~		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11237							
11238	043464	005037	001662	CLR	INTSET		;CLEAR INTERRUPT FLAG
11239	043470	104437		TWAT8S			;WAIT FOR SECOND INTERRUPT
11240	043472	104002		ERROR	2		
11241							
11242	043474	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11243	043476	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11244							
11245	043500	104416		TSSINIT			;CLEAR SUBSYSTEM
11246	043502	104003		ERROR	3		;BAD INIT ERROR
11247							
11248	043504	005737	001676	TST	REFMT		;TEST IF REFORMAT REQUIRED
11249	043510	001435		BEQ	PROGEND		;NO - GO TO HALT
11250	043512	104401	053013	TYPE	,OPR020		;TYPE MESSAGE
11251							
11252	043516	004437	035530	JSR	R4,LRLoad		;LOAD "L" REGS
11253	043522	000127		WRHEAD			;WRHEAD
11254	043524	177676		-102			; -102 WORDS
11255	043526	063414		OBUFF			;OBUFF IS BUFF ADDRESS
11256	043530	000		.BYTE	0		;SECTOR 0
11257	043531	000		.BYTE	0		;TRACK 0
11258	043532	000312			312		;CYLINDER 312
11259							
11260	043534	005737	001676	TST	REFMT		;TEST IF CYL 0
11261	043540	100002		BPL	5\$		;NO - SKIP
11262	043542	005037	001614	CLR	L.DCYL		;ELSE LOAD FOR CYL 0
11263	043546	004437	041740	JSR	R4,GENCOM		;GENERATE HEADERS
11264	043552	001200			1200		
11265							
11266	043554	104417		TLOADRK			;LOAD RK REGS
11267	043556	104434		TWAT159			;WAIT FOR INTERRUPT
11268	043560	104002		ERROR	2		;TO SLOW/NOT COMPLETE ERROR
11269							
11270	043562	104421		TCHKOP			;CHECK OPERATION FOR ANY ERRORS
11271	043564	104004		ERROR	4 ;OR 5, 6, 7		;REPORT ALL ERRORS
11272							
11273	043566	122737	000002 001607	CMPB	#2,L.DT		;TEST IF TRACK 2 FORMATTED
11274	043574	001403		BEQ	PROGEND		;YES - SKIP
11275	043576	105237	001607	INCB	L.DT		;ELSE BUMP TRACK
11276	043602	000761		BR	5\$		;DO NEXT TRACK
11277							
11278	043604	104401	053107	PROGEND:	TYPE OPR021		;TYPE HALT MESSAGE
11279	043610	012706	001100	MOV	#STACK,SP		;CLEAR STACK
11280	043614	105037	001103	CLRB	SEFLG		;CLEAR ERROR FLAG
11281	043620	005037	001264	CLR	SESCAPE		;CLEAR ESCAPE
11282	043624	005737	000042	TST	#42		;TEST IF MONITOR PRESENT
11283	043630	001404		BEQ	10\$		;NO - SKIP

C01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 208
CONTROLLED HALT SUBROUTINE

SEQ 0208

```

11284 043632 005037 032016          CLR      SEOPCT      ;SET FOR END OF PROGRAM
11285 043636 000137 031770          JMP      SEOP      ;GO TO END OF PASS
11286
11287 043642 000000          10$:  HALT      ;HALT PROGRAM
11288 043644 000137 001754          JMP      START1    ;GO TO RESTART IF CONTINUE
11289
11290 .SBTTL  HALT FAIL ROUTINE
11291 ;*      THIS ROUTINE IS ENTERED IF A HARDWARE ERROR IS DETECTED WHEN
11292 ;*      THE CARTRIDGE IS BEING REFORMATTED PRIOR TO HALT.
11293 043650 000240          ABTFAIL:  NOP
11294 043652 104401 053150          TYPE      OPRQ22      ;TYPE HALT FAIL MESSAGE
11295 043656 000137 043604          JMP      PROGEND    ;GO STOP PROGRAM
11296
11297 .SBTTL  TYPE ROUTINE
11298
11299 ;*****
11300 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
11301 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
11302 ;NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
11303 ;NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
11304 ;NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
11305 ;*
11306 ;CALL:
11307 ;*1) USING A TRAP INSTRUCTION
11308 ;*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
11309 ;*OR
11310 ;*      TYPE
11311 ;*      MESADR
11312 ;*
11313 043662 105737 001157          $TYPE:  TSTB      STPFLG      ;IS THERE A TERMINAL?
11314 043666 100002          BPL      1$      ;BR IF YES
11315 043670 000000          HALT      ;HALT HERE IF NO TERMINAL
11316 043672 000430          BR      3$      ;LEAVE
11317 043674 010046          1$:  MOV      R0,-(SP)    ;SAVE R0
11318 043676 017600 000002          MOV      22(SP),R0    ;GET ADDRESS OF ASCIZ STRING
11319 043702 122737 000001 001316          CMPB      #APTENV,$ENV    ;RUNNING IN APT MODE
11320 043710 001011          BNE      62$      ;NO,GO CHECK FOR APT CONSOLE
11321 043712 132737 000100 001317          BITB      #APTSPool,$ENVM    ;SPOOL MESSAGE TO APT
11322 043720 001405          BEQ      62$      ;NO,GO CHECK FOR CONSOLE
11323 043722 010037 043732          MOV      R0,61$      ;SETUP MESSAGE ADDRESS FOR APT
11324 043726 004737 033266          JSR      PC,$ATY3    ;SPOOL MESSAGE TO APT
11325 043732 000000          61$:  .WORD      0      ;MESSAGE ADDRESS
11326 043734 132737 000040 001317          62$:  BITB      #APTCSUP,$ENVM    ;APT CONSOLE SUPPRESSED
11327 043742 001003          BNE      60$      ;YES,SKIP TYPE OUT
11328 043744 112046          2$:  MOVB      (R0)+,-(SP)    ;PUSH CHARACTER TO BE TYPED ONTO STACK
11329 043746 001005          BNE      4$      ;BR IF IT ISN'T THE TERMINATOR
11330 043750 005726          TST      (SP)+      ;IF TERMINATOR POP IT OFF THE STACK
11331 043752 012600          60$:  MOV      (SP)+,R0    ;RESTORE R0
11332 043754 062716 000002          3$:  ADD      #2,(SP)    ;ADJUST RETURN PC
11333 043760 000002          RTI      ;RETURN
11334 043762 122716 000011          4$:  CMPB      #HT,(SP)    ;BRANCH IF <HT>
11335 043766 001430          BEQ      8$      ;
11336 043770 122716 000200          CMPB      #CRLF,(SP)    ;BRANCH IF NOT <CRLF>
11337 043774 001006          BNE      5$      ;
11338 043776 005726          TST      (SP)+      ;POP <CR><LF> EQUIV
11339 044000 104401          TYPE      ;TYPE A CR AND LF

```

001

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 209
TYPE ROUTINE

SEQ 0209

```

11340 044002 001273          SCRLF
11341 044004 105037 044140  CLRB    $CHARCNT      ;; CLEAR CHARACTER COUNT
11342 044010 000755          BR      2$          ;; GET NEXT CHARACTER
11343 044012 004737 044074  5$: JSR    PC,$TYPEC    ;; GO TYPE THIS CHARACTER
11344 044016 123726 001156  6$: CMPB  $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
11345 044022 001350          BNE    2$          ;; IF NO GO GET NEXT CHAR.
11346 044024 013746 001154  MOV     $NULL,-(SP)  ;; GET # OF FILLER CHARS. NEEDED
11347                                AND THE NULL CHAR.
11348 044030 105366 000001  7$: DECB  1(SP)      ;; DOES A NULL NEED TO BE TYPED?
11349 044034 002770          BLT    6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK
11350 044036 004737 044074  JSR    PC,$TYPEC    ;; GO TYPE A NULL
11351 044042 105337 044140  DECB  $CHARCNT      ;; DO NOT COUNT AS A COUNT
11352 044046 000770          BR      7$          ;; LOOP
11353
11354                                ;HORIZONTAL TAB PROCESSOR
11355
11356 044050 112716 000040  8$: MOVB  #' (SP)      ;; REPLACE TAB WITH SPACE
11357 044054 004737 044074  9$: JSR    PC,$TYPEC    ;; TYPE A SPACE
11358 044060 132737 000007 044140 BITB  #7,$CHARCNT  ;; BRANCH IF NOT AT
11359 044066 001372          BNE    9$          ;; TAB STOP
11360 044070 005726          TST    (SP)+        ;; POP SPACE OFF STACK
11361 044072 000724          BR      2$          ;; GET NEXT CHARACTER
11362 044074 105777 135050  $TYPEC: TSTB  0$TPS      ;; WAIT UNTIL PRINTER IS READY
11363 044100 100375          BPL    $TYPEC
11364 044102 116677 000002 135042 MOVB  2(SP),0$TPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
11365 044110 122766 000015 000002 CMPB  #CR,2(SP)    ;; IS CHARACTER A CARRIAGE RETURN?
11366 044116 001003          BNE    1$          ;; BRANCH IF NO
11367 044120 105037 044140  CLRB  $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
11368 044124 000406          BR      $TYPEX      ;; EXIT
11369 044126 122766 000012 000002 1$: CMPB  #LF,2(SP)  ;; IS CHARACTER A LINE FEED?
11370 044134 001402          BEQ    $TYPEX      ;; BRANCH IF YES
11371 044136 105227          INCB  (PC)+        ;; COUNT THE CHARACTER
11372 044140 000000          $CHARCNT: .WORD 0      ;; CHARACTER COUNT STORAGE
11373 044142 000207          $TYPEX: RTS    PC
11374
11375                                .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
11376
11377                                ;*****
11378                                ;*THIS ROUTINE IS USED TO CHANGE A 16 BIT BINARY NUMBER TO A 5-DIGIT
11379                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
11380                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
11381                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
11382                                ;*REPLACED WITH SPACES.
11383                                ;*CALL:
11384                                ;*      MOV     NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
11385                                ;*      TYPDS      ;; GO TO THE ROUTINE
11386
11387                                $TYPDS:
11388 044144 010046          MOV     R0,-(SP)      ;; PUSH R0 ON STACK
11389 044146 010146          MOV     R1,-(SP)      ;; PUSH R1 ON STACK
11390 044150 010246          MOV     R2,-(SP)      ;; PUSH R2 ON STACK
11391 044152 010346          MOV     R3,-(SP)      ;; PUSH R3 ON STACK
11392 044154 010546          MOV     R5,-(SP)      ;; PUSH R5 ON STACK
11393 044156 012746 020200  MOV     #20200,-(SP)  ;; SET BLANK SWITCH AND SIGN
11394 044162 016605 000020  MOV     20(SP),R5    ;; GET THE INPUT NUMBER
11395 044166 100004          BPL    1$          ;; BR IF INPUT IS POS.

```

EO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 210
CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

SEQ 0210

```

11396 044170 005405          NEG      R5          ;; MAKE THE BINARY NUMBER POS.
11397 044172 112766 000055 000001 1$:  MOVB    #'-,1(SP)  ;; MAKE THE ASCII NUMBER NEG.
11398 044200 005000          CLR      R0          ;; ZERO THE CONSTANTS INDEX
11399 044202 012703 044360          MOV     #$DBLK,R3  ;; SETUP THE OUTPUT POINTER
11400 044206 112723 000040          MOVB    #' ,(R3)+  ;; SET THE FIRST CHARACTER TO A BLANK
11401 044212 005002          CLR      R2          ;; CLEAR THE BCD NUMBER
11402 044214 016001 044350          MOV     $DTBL(R0),R1  ;; GET THE CONSTANT
11403 044220 160105          3$:  SUB      R1,R5      ;; FORM THIS BCD DIGIT
11404 044222 002402          BLT      4$          ;; BR IF DONE
11405 044224 005202          INC      R2          ;; INCREASE THE BCD DIGIT BY 1
11406 044226 000774          BR       3$
11407 044230 060105          4$:  ADD      R1,R5      ;; ADD BACK THE CONSTANT
11408 044232 005702          TST      R2          ;; CHECK IF BCD DIGIT=0
11409 044234 001002          BNE      5$          ;; FALL THROUGH IF 0
11410 044236 105716          TSTB     (SP)          ;; STILL DOING LEADING 0'S?
11411 044240 100407          BMI      7$          ;; BR IF YES
11412 044242 106316          5$:  ASLB     (SP)          ;; MSD?
11413 044244 103003          BCC      6$          ;; BR IF NO
11414 044246 116663 000001 177777 6$:  MOVB    1(SP),-1(R3)  ;; YES--SET THE SIGN
11415 044254 052702 000060          BIS     #'0,R2      ;; MAKE THE BCD DIGIT ASCII
11416 044260 052702 000040          7$:  BIS     #' ,R2      ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
11417 044264 110223          MOVB    R2,(R3)+  ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
11418 044266 005720          TST      (R0)+  ;; JUST INCREMENTING
11419 044270 020027 000010          CMP     R0,#10      ;; CHECK THE TABLE INDEX
11420 044274 002746          BLT      2$          ;; GO DO THE NEXT DIGIT
11421 044276 003002          BGT      8$          ;; GO TO EXIT
11422 044300 010502          MOV     R5,R2          ;; GET THE LSD
11423 044302 000764          BR       6$          ;; GO CHANGE TO ASCII
11424 044304 105726          8$:  TSTB     (SP)+  ;; WAS THE LSD THE FIRST NON-ZERO?
11425 044306 100003          BPL      9$          ;; BR IF NO
11426 044310 116663 177777 177776 9$:  MOVB    -1(SP),-2(R3)  ;; YES--SET THE SIGN FOR TYPING
11427 044316 105013          CLRB     (R3)          ;; SET THE TERMINATOR
11428 044320 012605          MOV     (SP)+,R5      ;; POP STACK INTO R5
11429 044322 012603          MOV     (SP)+,R3      ;; POP STACK INTO R3
11430 044324 012602          MOV     (SP)+,R2      ;; POP STACK INTO R2
11431 044326 012601          MOV     (SP)+,R1      ;; POP STACK INTO R1
11432 044330 012600          MOV     (SP)+,R0      ;; POP STACK INTO R0
11433 044332 104401 044360          TYPE     $DBLK      ;; NOW TYPE THE NUMBER
11434 044336 016666 000002 000004  MOV     2(SP),4(SP)  ;; ADJUST THE STACK
11435 044344 012616          MOV     (SP)+,(SP)
11436 044346 000002          RTI
11437 044350 023420          $DTBL: 10000.          ;; RETURN TO USER
11438 044352 001750          1000.
11439 044354 000144          100.
11440 044356 000012          10.
11441 044360 000004          $DBLK: .BLKW 4
11442          .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
11443          *****
11444          *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
11445          *OCTAL (ASCII) NUMBER AND TYPE IT.
11446          *STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
11447          *CALL:
11448          *      MOV     NUM,-(SP)          ;; NUMBER TO BE TYPED
11449          *      TYPOS          ;; CALL FOR TYPEOUT
11450          *      .EYTE  N          ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
11451

```

F01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 211
BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0211

```

11452      *      .BYTE      M      ;;M=1 OR 0
11453      *
11454      *      ;;1=TYPE LEADING ZEROS
11455      *      ;;0=SUPPRESS LEADING ZEROS
11456      *STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
11457      *STYPOS OR STYPOC
11458      *CALL:
11459      *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
11460      *      TYPON      ;;CALL FOR TYPEOUT
11461      *
11462      *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
11463      *CALL:
11464      *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
11465      *      TYPOC      ;;CALL FOR TYPEOUT
11466      *
11467      044370      017646      000000      STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
11468      044374      116637      000001      044613      MOV      1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
11469      044402      112637      044615      MOV      (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
11470      044406      062716      000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
11471      044412      000406      BR      STYPON
11472      044414      112737      000001      044613      STYPOC: MOV      #1,SOFILL      ;;SET THE ZERO FILL SWITCH
11473      044422      112737      000006      044615      MOV      #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
11474      044430      112737      000005      044612      STYPON: MOV      #5,SOCNT      ;;SET THE ITERATION COUNT
11475      044436      010346      MOV      R3,-(SP)      ;;SAVE R3
11476      044440      010446      MOV      R4,-(SP)      ;;SAVE R4
11477      044442      010546      MOV      R5,-(SP)      ;;SAVE R5
11478      044444      113704      044615      MOV      SOMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
11479      044450      005404      NEG      R4
11480      044452      062704      000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
11481      044456      110437      044614      MOV      R4,SOMODE      ;;SAVE IT FOR USE
11482      044462      113704      044613      MOV      SOFILL,R4      ;;GET THE ZERO FILL SWITCH
11483      044466      016605      000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
11484      044472      005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
11485      044474      006105      1$:      ROL      R5      ;;ROTATE MSB INTO "C"
11486      044476      000404      BR      3$      ;;GO DO MSB
11487      044500      006105      2$:      ROL      R5      ;;FORM THIS JIGIT
11488      044502      006105      ROL      R5
11489      044504      006105      ROL      R5
11490      044506      010503      MOV      R5,R3
11491      044510      006103      3$:      ROL      R3      ;;GET LSB OF THIS DIGIT
11492      044512      105337      044614      DECB      SOMODE      ;;TYPE THIS DIGIT?
11493      044516      100016      BPL      7$      ;;BR IF NO
11494      044520      042703      177770      BIC      #177770,R3      ;;GET RID OF JUNK
11495      044524      001002      BNE      4$      ;;TEST FOR 0
11496      044526      005704      TST      R4      ;;SUPPRESS THIS 0?
11497      044530      001403      BEQ      5$      ;;BR IF YES
11498      044532      005204      4$:      INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
11499      044534      052703      000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
11500      044540      052703      000040      5$:      BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
11501      044544      110337      044610      MOV      R3,R5      ;;SAVE FOR TYPING
11502      044550      104401      044610      TYPE      8$      ;;GO TYPE THIS DIGIT
11503      044554      105337      044612      7$:      DECB      SOCNT      ;;COUNT BY 1
11504      044560      003347      BGT      2$      ;;BR IF MORE TO DO
11505      044562      002402      BLT      6$      ;;BR IF DONE
11506      044564      005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
11507      044566      000744      BR      2$      ;;GO DO THE LAST DIGIT

```

GO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 212
BINARY TO OCTAL (ASCII) AND TYPE

SEQ 0212

11508	044570	012605			65:	MOV	(SP)+,R5	::RESTORE R5
11509	044572	012604				MOV	(SP)+,R4	::RESTORE R4
11510	044574	012603				MOV	(SP)+,R3	::RESTORE R3
11511	044576	016666	000002	000004		MOV	2(SP),4(SP)	::SET THE STACK FOR RETURNING
11512	044604	012616				MOV	(SP)+,(SP)	
11513	044606	000002				RTI		::RETURN
11514	044610	000			85:	.BYTE	0	::STORAGE FOR ASCII DIGIT
11515	044611	000				.BYTE	0	::TERMINATOR FOR TYPE ROUTINE
11516	044612	000			\$OCNT:	.BYTE	0	::OCTAL DIGIT COUNTER
11517	044613	000			\$OFILL:	.BYTE	0	::ZERO FILL SWITCH
11518	044614	000000			\$OMODE:	.WORD	0	::NUMBER OF DIGITS TO TYPE
11519					.SBTTL	TTY INPUT ROUTINE		
11520					*****			
11521					.ENABL	LSB		
11522					\$TKCNT:	.WORD	0	::NUMBER OF ITEMS IN QUEUE
11523	044616	000000			\$TKQIN:	.WORD	0	::INPUT POINTER
11524	044620	000000			\$TKQOUT:	.WORD	0	::OUTPUT POINTER
11525	044622	000000			\$TKQSRV:	.BLKB	1	::TTY KEYBOARD QUEUE
11526	044624	000001			\$TKQEND=.			
11527		044625			.EVEN			
11528		044626						
11529								
11530					*TK INITIALIZE ROUTINE			
11531					*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE			
11532					*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT			
11533								
11534					*CALL:			
11535					*JSR	PC,\$TKINT		
11536					*RETURN			
11537								
11538	044626	005037	044616		\$TKINT:	CLR	\$TKCNT	::CLEAR COUNT OF ITEMS IN QUEUE
11539	044632	012737	044624	044620		MOV	\$TKQSRV,\$TKQIN	::MOVE THE STARTING ADDRESS OF THE
11540	044640	013737	044620	044622		MOV	\$TKQIN,\$TKQOUT	::QUEUE INTO THE INPUT & OUTPUT POINTERS.
11541	044646	012737	044676	000060		MOV	\$TKSRV,\$TKVEC	::INITIALIZE THE KEYBOARD VECTOR
11542	044654	012737	000200	000062		MOV	\$200,\$TKVEC+2	::"BR" LEVEL 4
11543	044662	005777	134260			TST	\$TKB	::CLEAR DONE FLAG
11544	044666	012777	000100	134250		MOV	\$100,\$TKS	::ENABLE TTY KEYBOARD INTERRUPT
11545	044674	000207				RTS	PC	::RETURN TO CALLER
11546								
11547					*TK SERVICE ROUTINE			
11548					*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT			
11549					*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING			
11550					*IT IN THE QUEUE.			
11551					*IF THE CHARACTER IS A "CONTROL-C" (↑C) \$TKINT IS CALLED AND			
11552					*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (CTRHLT)			
11553								
11554	044676	117746	134244		\$TKSRV:	MOVB	\$TKB,-(SP)	::PICKUP THE CHARACTER
11555	044702	042716	177600			BIC	\$↑C177,(SP)	::STRIP THE JUNK
11556	044706	021627	000003			CMP	(SP),#3	::IS IT A CONTROL C?
11557	044712	001007				BNE	15	::BRANCH IF NO
11558	044714	104401	046012			TYPE	\$CNTLC	::TYPE A CONTROL-C (↑C)
11559	044720	004737	044626			JSR	PC,\$TKINT	::INIT THE KEYBOARD
11560	044724	005726				TST	(SP)+	::CLEAN UP STACK
11561	044726	000137	043404			JMP	CTRHLT	::CONTROL C RESTART
11562	044732	021627	000007		15:	CMP	(SP),#7	::IS IT A CONTROL G?
11563	044736	001004				BNE	25	::BRANCH IF NO

H01

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR&KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 213
TTY INPUT ROUTINE

SEQ 0213

```

11564 044740 022737 000176 001140      CMP      #SWREG,SWR      ;; IS SOFT-SWR SELECTED?
11565 044746 001500                      BEQ      6$              ;; GO TO SWR CHANGE
11566
11567 044750                      2$:      CMP      #1,$STKCNT      ;; IS THE QUEUE FULL?
11568 044750 022737 000001 044616          BNE      3$              ;; BRANCH IF NO
11569 044756 001004                      TYPE      $BELL      ;; RING THE TTY BELL
11570 044760 104401 001266                  TST      (SP)+      ;; CLEAN CHARACTER OFF OF STACK
11571 044764 005726                      BR        5$              ;; EXIT
11572 044766 000451                      3$:      CMP      (SP),#23      ;; IS IT A CONTROL-S?
11573 044770 021627 000023                  BNE      32$             ;; BRANCH IF NO
11574 044774 001021                      CLR      2$TKS      ;; DISABLE TTY KEYBOARD INTERRUPTS
11575 044776 005077 134142                  TST      (SP)+      ;; CLEAN CHAR OFF STACK
11576 045002 005726                      31$:     TSTB     2$TKS      ;; WAIT FOR A CHAR
11577 045004 105777 134134                  BPL      31$          ;; LOOP UNTIL ITS THERE
11578 045010 100375                      MOV      2$TKB, -(SP)    ;; GET THE CHARACTER
11579 045012 117746 134130                  BIC      #1C177, (SP)  ;; MAKE IT 7-BIT ASCII
11580 045016 042716 177600                  CMP      (SP)+, #21    ;; IS IT A CONTROL-Q?
11581 045022 022627 000021                  BNE      31$          ;; BRANCH IF NO
11582 045026 001366                      MOV      #100, 2$TKS  ;; REENABLE TTY KEYBOARD INTERRUPTS
11583 045030 012777 000100 134106          RTI              ;; RETURN
11584 045036 000002                      32$:     INC      $TKCNT      ;; COUNT THIS CHARACTER
11585 045040 005237 044616                  CMP      (SP), #140    ;; IS IT UPPER CASE?
11586 045044 021627 000140                  BLT      4$              ;; BRANCH IF YES
11587 045050 002405                      CMP      (SP), #175    ;; IS IT A SPECIAL CHAR?
11588 045052 021627 000175                  BGT      4$              ;; BRANCH IF YES
11589 045056 003002                      BIC      #40, (SP)     ;; MAKE IT UPPER CASE
11590 045060 042716 000040                  MOV      (SP)+, 2$TKQIN  ;; AND PUT IT IN QUEUE
11591 045064 112677 177530                  INC      $TKQIN        ;; UPDATE THE POINTER
11592 045070 005237 044620                  CMP      $TKQIN, $STKQEND  ;; GO OFF THE END?
11593 045074 023727 044620 044625          BNE      5$              ;; BRANCH IF NO
11594 045102 001003                      MOV      $STKQSR, $TKQIN  ;; RESET THE POINTER
11595 045104 012737 044624 044620          5$:      RTI              ;; RETURN
11596 045112 000002
11597
11598      ;; *****
11599      ;; *SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
11600      ;; *ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
11601      ;; *SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
11602      ;; *CALL WHEN OPERATING IN TTY INTERRUPT MODE.
11603 045114 022737 000176 001140 $CKSWR: CMP      #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED
11604 045122 001124                      BNE      15$          ;; EXIT IF NOT
11605 045124 105777 134014                  TSTB     2$TKS      ;; IS A CHAR WAITING?
11606 045130 100121                      BPL      15$          ;; IF NOT, EXIT
11607 045132 117746 134010                  MOV      2$TKB, -(SP)    ;; YES
11608 045136 042716 177600                  BIC      #1C177, (SP)  ;; MAKE IT 7-BIT ASCII
11609 045142 021627 000007                  CMP      (SP), #7      ;; IS IT A CONTROL-G?
11610 045146 001300                      BNE      2$              ;; IF NOT, PUT IT IN THE TTY QUEUE
11611
11612                      ;; AND EXIT
11613
11614      ;; *****
11615      ;; *CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
11616      ;; *ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
11617      ;; *CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
11618 045150 123727 001134 000001 6$:      CMPB     $AUTOB, #1      ;; ARE WE RUNNING IN AUTO-MODE?
11619 045156 001674                      BEQ      2$              ;; BRANCH IF YES
11619 045160 005726                      TST      (SP)+      ;; CLEAR CONTROL-G OFF STACK

```

11620	045162	004737	044626		JSR	PC, STKINT	:: FLUSH THE TTY INPUT QUEUE
11621	045166	005077	133752		CLR	2STKS	:: DISABLE TTY KEYBOARD INTERRUPTS
11622	045172	112737	000001	001135	MOVB	#1, \$INTAG	:: SET INTERRUPT MODE INDICATOR
11623							
11624	045200	104401	046024		TYPE	, \$CNTLG	:: ECHO THE CONTROL-G (↑G)
11625	045204	104401	046031		SGTSWR: TYPE	, \$MSWR	:: TYPE CURRENT CONTENTS
11626	045210	013746	000176		MOV	\$WREG, -(SP)	:: SAVE \$WREG FOR TYPEOUT
11627	045214	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
11628	045216	104401	046042		TYPE	, \$MNEW	:: PROMPT FOR NEW SWR
11629	045222	005046			19\$: CLR	-(SP)	:: CLEAR COUNTER
11630	045224	005046			CLR	-(SP)	:: THE NEW SWR
11631	045226	105777	133712		7\$: TSTB	2STKS	:: CHAR THERE?
11632	045232	100375			BPL	7\$	:: IF NOT TRY AGAIN
11633							
11634	045234	117746	133706		MOVB	2STKB, -(SP)	:: PICK UP CHAR
11635	045240	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
11636							
11637	045244	021627	000003		CMP	(SP), #3	:: IS IT A CONTROL-C?
11638	045250	001015			BNE	9\$	:: BRANCH IF NOT
11639	045252	104401	046012		TYPE	, \$CNTLC	:: YES, ECHO CONTROL-C (↑C)
11640	045256	062706	000006		ADD	#6, SP	:: CLEAN UP STACK
11641	045262	123727	001135	000001	CMPB	\$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
11642	045270	001003			BNE	8\$	:: BRANCH IF NO
11643	045272	012777	000100	133644	MOV	#100, 2STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
11644	045300	000137	043404		8\$: JMP	CTRHLT	:: CONTROL-C RESTART
11645							
11646							
11647	045304	021627	000025		9\$: CMP	(SP), #25	:: IS IT A CONTROL-U?
11648	045310	001005			BNE	10\$	:: BRANCH IF NOT
11649	045312	104401	046017		TYPE	, \$CNTLU	:: YES, ECHO CONTROL-U (↑U)
11650	045316	062706	000006		20\$: ADD	#6, SP	:: IGNORE PREVIOUS INPUT
11651	045322	000737			BR	19\$	:: LET'S TRY IT AGAIN
11652							
11653							
11654	045324	021627	000015		10\$: CMP	(SP), #15	:: IS IT A <CR>?
11655	045330	001022			BNE	16\$	:: BRANCH IF NO
11656	045332	005766	000004		TST	4(SP)	:: YES, IS IT THE FIRST CHAR?
11657	045336	001403			BEQ	11\$	:: BRANCH IF YES
11658	045340	016677	000002	133572	MOV	2(SP), 2SWR	:: SAVE NEW SWR
11659	045346	062706	000006		11\$: ADD	#6, SP	:: CLEAN UP STACK
11660	045352	104401	001273		14\$: TYPE	, \$CRLF	:: ECHO <CR> AND <LF>
11661	045356	123727	001135	000001	CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
11662	045364	001003			BNE	15\$	:: BRANCH IF NOT
11663	045366	012777	000100	133550	MOV	#100, 2STKS	:: RE-ENABLE TTY KBD INTERRUPTS
11664	045374	000002			15\$: RTI		:: RETURN
11665	045376	004737	044074		16\$: JSR	PC, \$TYPEC	:: ECHO CHAR
11666	045402	021627	000060		CMP	(SP), #60	:: CHAR < 0?
11667	045406	002420			BLT	18\$	:: BRANCH IF YES
11668	045410	021627	000067		CMP	(SP), #67	:: CHAR > 7?
11669	045414	003015			BGT	18\$	:: BRANCH IF YES
11670	045416	042726	000060		BIC	#60, (SP)+	:: STRIP-OFF ASCII
11671	045422	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
11672	045426	001403			BEQ	17\$	:: BRANCH IF YES
11673	045430	006316			ASL	(SP)	:: NO, SHIFT PRESENT
11674	045432	006316			ASL	(SP)	:: CHAR OVER TO MAKE
11675	045434	006316			ASL	(SP)	:: ROOM FOR NEW ONE.

J01

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZR6KD.P11 31-JAN-77 16:26

 MACY11 27(1006) 31-JAN-77 16:34 PAGE 215
 TTY INPUT ROUTINE

SEG 0215

11676	045436	005266	000002		17\$: INC 2(SP)	;; KEEP COUNT OF CHAR
11677	045442	056616	177776		BIS -2(SP), (SP)	;; SET IN NEW CHAR
11678	045446	000667			BR 7\$	;; GET THE NEXT ONE
11679	045450	104401	001272		18\$: TYPE \$QUES	;; TYPE ?(CR)(LF)
11680	045454	000720			BR 20\$	;; SIMULATE CONTROL-U
11681					.DSABL LSB	
11682						
11683						
11684						
11685					*****	
11686					;; THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY	
11687					;; *CALL:	
11688					RDCHR	;; GET A CHARACTER FROM THE QUEUE
11689					RETURN HERE	;; CHARACTER IS ON THE STACK
11690						;; WITH PARITY BIT STRIPPED OFF
11691						
11692	045456	011646			\$RDCHR: MOV (SP), -(SP)	;; PUSH DOWN THE PC AND
11693	045460	016666	000004	000002	MOV 4(SP), 2(SP)	;; THE PS
11694	045466	005066	000004		CLR 4(SP)	;; GET READY FOR A CHARACTER
11695	045472	005046			CLR -(SP)	;; PUT NEW PS ON STACK
11696	045474	012746	045502		MOV #64\$, -(SP)	;; PUT NEW PC ON STACK
11697	045500	000002			RTI	;; POP NEW PC AND PS
11698	045502					
11699	045502	005737	044616		64\$: TST \$TKCNT	;; WAIT ON A CHARACTER
11700	045506	001775			1\$ BEQ 1\$	
11701	045510	005337	044616		DEC \$TKCNT	;; DECREMENT THE COUNTER
11702	045514	117766	177102	000004	MOVB \$TKQOUT, 4(SP)	;; GET ONE CHARACTER
11703	045522	005237	044622		INC \$TKQOUT	;; UPDATE THE POINTER
11704	045526	023727	044622	044625	CMP \$TKQOUT, \$TKQEND	;; DID IT GO OFF OF THE END?
11705	045534	001003			BNE 2\$	;; BRANCH IF NO
11706	045536	012737	044624	044622	MOV \$TKQSRT, \$TKQOUT	;; RESET THE POINTER
11707	045544	000002			2\$ RTI	;; RETURN
11708					*****	
11709					;; THIS ROUTINE WILL INPUT A STRING FROM THE TTY	
11710					;; *CALL:	
11711					RDLIN	;; INPUT A STRING FROM THE TTY
11712					RETURN HERE	;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
11713						;; TERMINATOR WILL BE A BYTE OF ALL 0'S
11714						
11715	045546	010346			\$RDLIN: MOV R3, -(SP)	;; SAVE R3
11716	045550	005046			CLR -(SP)	;; CLEAR THE RUBOUT KEY
11717	045552	012703	046002		1\$: MOV \$TTYIN, R3	;; GET ADDRESS
11718	045556	022703	046012		2\$: CMP \$TTYIN+8., R3	;; BUFFER FULL?
11719	045562	101456			BLOS 4\$	;; BR IF YES
11720	045564	104410			RDCHR	;; GO READ ONE CHARACTER FROM THE TTY
11721	045566	112613			MOVB (SP)+, (R3)	;; GET CHARACTER
11722	045570	122713	000177		10\$: CMPB #177, (R3)	;; IS IT A RUBOUT
11723	045574	001022			BNE 5\$	;; BR IF NO
11724	045576	005716			TST (SP)	;; IS THIS THE FIRST RUBOUT?
11725	045600	001007			BNE 6\$	;; BR IF NO
11726	045602	112737	000134	046000	MOVB #'\, 9\$	;; TYPE A BACK SLASH
11727	045610	104401	046000		TYPE 9\$	
11728	045614	012716	177777		MOV #-1, (SP)	;; SET THE RUBOUT KEY
11729	045620	005303			6\$: DEC R3	;; BACKUP BY ONE
11730	045622	020327	046002		CMP R3, \$TTYIN	;; STACK EMPTY?
11731	045626	103434			BLO 4\$	;; BR IF YES

Address	Instruction	Comment
11732	045630 111337 046000	MOVW (R3), 95
11733	045634 104401 046000	TYPE 95
11734	045640 000746	BR 25
11735	045642 005716	55: TST (SP)
11736	045644 001406	BEQ 75
11737	045646 112737 000134 046000	MOVW #'\, 95
11738	045654 104401 046000	TYPE 95
11739	045660 005016	CLR (SP)
11740	045662 122713 000025	75: CMPB #25, (R3)
11741	045666 001003	BNE 85
11742	045670 104401 046017	TYPE \$CNTLU
11743	045674 000726	BR 15
11744	045676 122713 000022	85: CMPB #22, (R3)
11745	045702 001011	BNE 35
11746	045704 105013	CLRB (R3)
11747	045706 104401 001273	TYPE \$CRLF
11748	045712 104401 046002	TYPE \$TTYIN
11749	045716 000717	BR 25
11750	045720 104401 001272	45: TYPE \$QUES
11751	045724 000712	BR 15
11752	045726 111337 046000	35: MOVW (R3), 95
11753	045732 104401 046000	TYPE 95
11754	045736 122723 000015	CMPB #15, (R3)+
11755	045742 001305	BNE 25
11756	045744 105063 177777	CLRB -1(R3)
11757	045750 104401 001274	TYPE \$LF
11758	045754 005726	TST (SP)+
11759	045756 012603	MOV (SP)+, R3
11760	045760 011646	MOV (SP), -(SP)
11761	045762 016666 000004 000002	MOV 4(SP), 2(SP)
11762	045770 012766 046002 000004	MOV #TTYIN, 4(SP)
11763	045776 000002	RTI
11764	046000 000	95: .BYTE 0
11765	046001 000	.BYTE 0
11766	046002 000010	\$TTYIN: .BLKB 8
11767	046012 041536 005015 000	\$CNTLC: .ASCIZ /C/<15><12>
11768	046017 136 006525 000012	\$CNTLU: .ASCIZ /U/<15><12>
11769	046024 043536 005015 000	\$CNTLG: .ASCIZ /G/<15><12>
11770	046031 015 051412 051127	\$MSWR: .ASCIZ <15><12>/SWR = /
11771	046036 036440 000040	
11772	046042 020040 042516 020127	\$MNEW: .ASCIZ / NEW = /
11773	046050 020075 000	
11774	046054	.EVEN
11775		.SBTTL READ AN OCTAL NUMBER FROM THE TTY
11776		
11777		*****
11778		*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
11779		*CHANGE IT TO BINARY.
11780		*THE INPUT CHARACTERS WILL BE CHECKED TO INSURE THEY ARE LEGAL
11781		*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
11782		*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
11783		*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
11784		*CALL:
11785		* RDOCT
11786		* RETURN HERE
11787		* READ AN OCTAL NUMBER
		LOW ORDER BITS ARE ON TOP OF THE STACK
		HIGH ORDER BITS ARE IN \$HI OCT

L01

Address	Hex	Dec	Label	Instruction	Comment
11788	046054	011646		\$RDOCT: MOV (SP), -(SP)	;; PROVIDE SPACE FOR THE
11789	046056	016666	000004 000002	MOV 4(SP), 2(SP)	;; INPUT NUMBER
11791	046064	010046		MOV R0, -(SP)	;; PUSH R0 ON STACK
11792	046066	010146		MOV R1, -(SP)	;; PUSH R1 ON STACK
11793	046070	010246		MOV R2, -(SP)	;; PUSH R2 ON STACK
11794	046072	104411	1\$:	RDLIN	;; READ AN ASCII LINE
11795	046074	012600		MOV (SP)+, R0	;; GET ADDRESS OF 1ST CHARACTER
11796	046076	010037	046202	MOV R0, 5\$	;; AND SAVE IT
11797	046102	005001		CLR R1	;; CLEAR DATA WORD
11798	046104	005002		CLP R2	
11799	046106	112046	2\$:	MOV (R0)+, -(SP)	;; PICKUP THIS CHARACTER
11800	046110	001420		BEQ 3\$	;; IF ZERO GET OUT
11801	046112	122716	000060	CMPB #0, (SP)	;; MAKE SURE THIS CHARACTER
11802	046116	003026		BGT 4\$	;; IS AN OCTAL DIGIT
11803	046120	122716	000067	CMPB #7, (SP)	
11804	046124	002423		BLT 4\$	
11805	046126	006301		ASL R1	;; *2
11806	046130	006102		ROL R2	
11807	046132	006301		ASL R1	;; *4
11808	046134	006102		ROL R2	
11809	046136	006301		ASL R1	;; *8
11810	046140	006102		ROL R2	
11811	046142	042716	177770	BIC #C7, (SP)	;; STRIP THE ASCII JUNK
11812	046146	062601		ADD (SP)+, R1	;; ADD IN THIS DIGIT
11813	046150	000756		BR 2\$	;; LOOP
11814	046152	005726	3\$:	TST (SP)+	;; CLEAN TERMINATOR FROM STACK
11815	046154	010166	000012	MOV R1, 12(SP)	;; SAVE THE RESULT
11816	046160	010237	046212	MOV R2, \$HIOCT	
11817	046164	012602		MOV (SP)+, R2	;; POP STACK INTO R2
11818	046166	012601		MOV (SP)+, R1	;; POP STACK INTO R1
11819	046170	012600		MOV (SP)+, R0	;; POP STACK INTO R0
11820	046172	000002		RTI	;; RETURN
11821	046174	005726	4\$:	TST (SP)+	;; CLEAN PARTIAL FROM STACK
11822	046176	105010		CLRB (R0)	;; SET A TERMINATOR
11823	046200	104401		TYPE	;; TYPE UP THRU THE BAD CHAR.
11824	046202	000000	5\$:	.WORD 0	
11825	046204	104401	001272	TYPE \$QUES	;; "?" "CR" & "LF"
11826	046210	000730		BR 1\$	;; TRY AGAIN
11827	046212	000000	\$HIOCT: .WORD 0		;; HIGH ORDER BITS GO HERE
11828			.SBTTL	SAVE AND RESTORE R0-R5 ROUTINES	
11829					
11830					
11831					
11832					
11833					
11834					
11835					
11836					
11837					
11838					
11839					
11840					
11841					
11842					
11843					

MO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 218
SAVE AND RESTORE R0-R5 ROUTINES

SEQ 0218

```

11844
11845 046214
11846 046214 010046
11847 046216 010146
11848 046220 010246
11849 046222 010346
11850 046224 010446
11851 046226 010546
11852 046230 016646 000022
11853 046234 016646 000022
11854 046240 016646 000022
11855 046244 016646 000022
11856 046250 000002
11857
11858
11859
11860
11861 046252
11862 046252 012666 000022
11863 046256 012666 000022
11864 046262 012666 000022
11865 046266 012666 000022
11866 046272 012605
11867 046274 012604
11868 046276 012603
11869 046300 012602
11870 046302 012601
11871 046304 012600
11872 046306 000002
11873
11874
11875
11876
11877 046310 012737 046450 000024
11878 046316 012737 000340 000026
11879 046324 010046
11880 046326 010146
11881 046330 010246
11882 046332 010346
11883 046334 010446
11884 046336 010546
11885 046340 017746 132574
11886 046344 010637 046454
11887 046350 012737 046362 000024
11888 046356 000000
11889 046360 000776
11890
11891
11892
11893 046362 012737 046450 000024
11894 046370 013706 046454
11895 046374 005037 046454
11896 046400 005237 046454
11897 046404 001375
11898 046406 012677 132526
11899 046412 012605

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

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

.SBTTL POWER DOWN AND UP ROUTINES

;*****
;POWER DOWN ROUTINE
$PWRDN: MOV $SILLUP,2$PWRVEC ;; SET FOR FAST UP
MOV $340,2$PWRVEC+2 ;; PRI0:7
MOV R0,-(SP) ;; PUSH R0 ON STACK
MOV R1,-(SP) ;; PUSH R1 ON STACK
MOV R2,-(SP) ;; PUSH R2 ON STACK
MOV R3,-(SP) ;; PUSH R3 ON STACK
MOV R4,-(SP) ;; PUSH R4 ON STACK
MOV R5,-(SP) ;; PUSH R5 ON STACK
MOV 2$SWR,-(SP) ;; PUSH 2$SWR ON STACK
MOV SP,$SAVR6 ;; SAVE SP
MOV $PWRUP,2$PWRVEC ;; SET UP VECTOR
HALT
BR -.2 ;; HANG UP

;*****
;POWER UP ROUTINE
$PWRUP: MOV $SILLUP,2$PWRVEC ;; SET FOR FAST DOWN
MOV $SAVR6,SP ;; GET SP
CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
IS: INC $SAVR6 ;; WAIT FOR THE INC
BNE IS OF WORD
MOV (SP)+,2$SWR ;; POP STACK INTO 2$SWR
MOV (SP)+,R5 ;; POP STACK INTO R5

```

NO1

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 219
POWER DOWN AND UP ROUTINES

SEQ 0219

```

11900 046414 012604      MOV      (SP)+,R4      ;; POP STACK INTO R4
11901 046416 012603      MOV      (SP)+,R3      ;; POP STACK INTO R3
11902 046420 012602      MOV      (SP)+,R2      ;; POP STACK INTO R2
11903 046422 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
11904 046424 012600      MOV      (SP)+,R0      ;; POP STACK INTO R0
11905 046426 012737 046310 000024      MOV      $SPWRON,$$PWAVEC  ;; SET UP THE POWER DOWN VECTOR
11906 046434 012737 000340 000026      MOV      $340,$$PWAVEC+2  ;; PRI0:7
11907 046442 104401      TYPE      $POWER      ;; REPORT THE POWER FAILURE
11908 046444 046456      $PWAVEC: .WORD $POWER      ;; POWER FAIL MESSAGE POINTER
11909 046446 000002      RTI
11910 046450 000000      $ILLUP: HALT      ;; THE POWER UP SEQUENCE WAS STARTED
11911 046452 000776      BR      .-2      ;; BEFORE THE POWER DOWN WAS COMPLETE
11912 046454 000000      $SAVR6: 0      ;; PUT THE SP HERE
11913 046456 005015 047520 042527      $POWER: .ASCIZ <15><12>"POWER"
11914 046464 000122
11915
11916      .EVEN
11917      .SBTTL TRAP DECODER
11918
11919      ;*****
11920      ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
11921      ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
11922      ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
11923      ;*GO TO THAT ROUTINE.
11924
11924 046466 010046      $TRAP: MOV      R0,-(SP)      ;; SAVE R0
11925 046470 016600 000002      MOV      2(SP),R0      ;; GET TRAP ADDRESS
11926 046474 005740      TST      -(R0)      ;; BACKUP BY 2
11927 046476 111000      MOVB     (R0),R0      ;; GET RIGHT BYTE OF TRAP
11928 046500 006300      ASL      R0      ;; POSITION FOR INDEXING
11929 046502 016000 046522      MOV      $TRPAD(R0),R0      ;; INDEX TO TABLE
11930 046506 000200      RTS      R0      ;; GO TO ROUTINE
11931
11932
11933      ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
11934
11935 046510 011646      $TRAP2: MOV      (SP),-(SP)      ;; MOVE THE PC DOWN
11936 046512 016666 000004 000002      MOV      4(SP),2(SP)      ;; MOVE THE PSW DOWN
11937 046520 000002      RTI      ;; RESTORE THE PSW
11938
11939      .SBTTL TRAP TABLE
11940
11941      ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
11942      ;*BY THE "TRAP" INSTRUCTION.
11943
11944      ;
11945      ROUTINE
11946      -----
11946 046522 046510      $TRPAD: .WORD $TRAP2
11947 046524 043662      $TYPE      ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
11948 046526 044414      $TYPOC     ;;CALL=TYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
11949 046530 044370      $TYPOS     ;;CALL=TYPOS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
11950 046532 044430      $TYPON     ;;CALL=TYPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAS" CALL)
11951 046534 044144      $TYPDS     ;;CALL=TYPDS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
11952
11953 046536 045204      $GTSWR      ;;CALL=GTSWR     TRAP+6(104406) GET SOFT-SWR SETTING
11954
11955 046540 045114      $CKSWR      ;;CALL=CKSWR     TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR

```

11956	046542	045456	\$ROCHR	CALL=ROCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE
11957	046544	045546	\$ROLIN	CALL=ROLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE
11958	046546	046054	\$ROOCT	CALL=ROOCT	TRAP+12(104412)	READ AN OCTAL NUMBER FROM TTY
11959	046550	046214	\$SAVREG	CALL=SAVREG	TRAP+13(104413)	SAVE RO-RS ROUTINE
11960	046552	046252	\$RESREG	CALL=RESREG	TRAP+14(104414)	RESTORE RO-RS ROUTINE
11961	046554	035056	\$SCOPI\$	CALL=SCOPI	TRAP+15(104415)	INTERNAL LOOP ON ERROR
11962	046556	036656	\$SSINIT	CALL=SSINIT	TRAP+16(104416)	INITIALIZE SUBSYSTEM
11963	046560	035570	\$LOADRK	CALL=TLOADRK	TRAP+17(104417)	LOAD RK611 FOR OPERATION
11964	046562	035652	\$GETRK	CALL=TGETRK	TRAP+20(104420)	GET RK611 REGISTERS
11965	046564	037642	\$CHKOP	CALL=TCHKOP	TRAP+21(104421)	CHECK OPERATION FOR ANY ERRORS
11966	046566	037610	\$CHKWE	CALL=TCHKWE	TRAP+22(104422)	CHECK OPERATION FOR EXPECTED ERRORS
11967	046570	035364	\$IWAT16	CALL=TWAT16	TRAP+23(104423)	WAIT 16 MS
11968	046572	035354	\$IWAT32	CALL=TWAT32	TRAP+24(104424)	WAIT 32 MS
11969	046574	035344	\$IWAT48	CALL=TWAT48	TRAP+25(104425)	WAIT 48 MS
11970	046576	035334	\$IWAT64	CALL=TWAT64	TRAP+26(104426)	WAIT 64 MS
11971	046600	035324	\$IWAT80	CALL=TWAT80	TRAP+27(104427)	WAIT 80 MS
11972	046602	035314	\$IWAT96	CALL=TWAT96	TRAP+30(104430)	WAIT 96 MS
11973	046604	035304	\$IWAT112	CALL=TWAT112	TRAP+31(104431)	WAIT 112 MS
11974	046606	035274	\$IWAT128	CALL=TWAT128	TRAP+32(104432)	WAIT 128 MS
11975	046610	035264	\$IWAT144	CALL=TWAT144	TRAP+33(104433)	WAIT 144 MS
11976	046612	035254	\$IWAT159	CALL=TWAT159	TRAP+34(104434)	WAIT 160 MS
11977	046614	035244	\$IWAT1\$	CALL=TWAT1\$	TRAP+35(104435)	WAIT FOR 1 SECOND
11978	046616	035234	\$IWAT2\$	CALL=TWAT2\$	TRAP+36(104436)	WAIT FOR 2 SECONDS
11979	046620	035214	\$IWAT8\$	CALL=TWAT8\$	TRAP+37(104437)	WAIT FOR 8 SECONDS
11980	046622	035224	\$IWAT1M	CALL=TWAT1M	TRAP+40(104440)	WAIT FOR 1 MIN
11981		000102				

\$TERM=-\$TRPAD

C02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 221
DATA PATTERNS

SEQ 0221

11982			.SBTTL DATA PATTERNS
11983			;DATA PATTERN 1
11984			; PATTERN IS ALL ZEROS
11985			
11986			;DATA PATTERN 2
11987			; HI-LO FREQ. MIX
11988	046624		PAT02:
11989	046624	177777	177777
11990	046626	177777	177777
11991	046630	177777	177777
11992	046632	052525	052525
11993	046634	052525	052525
11994	046636	052525	052525
11995	046640	177777	177777
11996	046642	177777	177777
11997	046644	052525	052525
11998	046646	052525	052525
11999	046650	177777	177777
12000	046652	052525	052525
12001	046654	177252	177252
12002	046656	177252	177252
12003	046660	172765	172765
12004	046662	172765	172765
12005			
12006			;DATA PATTERN 3
12007			; HI FREQ. PHASE MIX
12008	046664		PAT03:
12009	046664	000000	000000
12010	046666	000000	000000
12011	046670	000000	000000
12012	046672	177777	177777
12013	046674	177777	177777
12014	046676	177777	177777
12015	046700	000000	000000
12016	046702	000000	000000
12017	046704	177777	177777
12018	046706	177777	177777
12019	046710	000000	000000
12020	046712	177777	177777
12021	046714	000000	000000
12022	046716	177777	177777
12023	046720	000000	000000
12024	046722	177777	177777
12025			
12026			;DATA PATTERN 4
12027			; LO FREQ. PHASE MIX
12028	046724		PAT04:
12029	046724	052525	052525
12030	046726	052525	052525
12031	046730	052525	052525
12032	046732	125252	125252
12033	046734	125252	125252
12034	046736	125252	125252
12035	046740	052525	052525
12036	046742	052525	052525
12037	046744	125252	125252

D02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 222
 DZR6KD.P11 31-JAN-77 16:26 DATA PATTERNS

SEQ 0222

12038	046746	125252	125252
12039	046750	052525	052525
12040	046752	125252	125252
12041	046754	052525	052525
12042	046756	125252	125252
12043	046760	052525	052525
12044	046762	125252	125252
12045			
12046			
12047			
12048	046764		
12049	046764	133333	133333
12050	046766	066666	066666
12051	046770	155555	155555
12052	046772	155555	155555
12053	046774	133333	133333
12054	046776	066666	066666
12055	047000	066666	066666
12056	047002	155555	155555
12057	047004	155555	155555
12058	047006	133333	133333
12059	047010	133333	133333
12060	047012	133333	133333
12061	047014	133333	133333
12062	047016	133333	133333
12063	047020	133333	133333
12064	047022	133333	133333
12065			
12066			
12067			
12068	047024		
12069	047024	121105	121105
12070	047026	150442	150442
12071	047030	064221	064221
12072	047032	132110	132110
12073	047034	055044	055044
12074	047036	026422	026422
12075	047040	013211	013211
12076	047042	105504	105504
12077	047044	042642	042642
12078	047046	021321	021321
12079	047050	110550	110550
12080	047052	044264	044264
12081	047054	022132	022132
12082	047056	011055	011055
12083	047060	104426	104426
12084	047062	042213	042213
12085			
12086			
12087			
12088			
12089			
12090			
12091	047064		
12092	047064	026455	026455
12093	047066	113226	113226

; DATA PATTERN 5
 :
 : MAX PRECOMP. PHASE MIX
 PAT05:

; DATA PATTERN 6
 :
 : ROTATING BOUNDARY PULSE PRECOMP.
 PAT06:

; DATA PATTERN 7
 :
 : FIELD OF ALL ONES

; DATA PATTERN 10
 :
 : ROTATING CELL PULSE PRECOMP.
 PAT10:

E02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006)
DATA PATTERNS

31-JAN-77 16:34 PAGE 223

SEQ 0223

12094	047070	045513	045513
12095	047072	122645	122645
12096	047074	151322	151322
12097	047076	064551	064551
12098	047100	132264	132264
12099	047102	055132	055132
12100	047104	026455	026455
12101	047106	113226	113226
12102	047110	045513	045513
12103	047112	122645	122645
12104	047114	151322	151322
12105	047116	064551	064551
12106	047120	132264	132264
12107	047122	055132	055132
12108			
12109			
12110			
12111	047124		
12112	047124	000001	000001
12113	047126	000002	000002
12114	047130	000004	000004
12115	047132	000010	000010
12116	047134	000020	000020
12117	047136	000040	000040
12118	047140	000100	000100
12119	047142	000200	000200
12120	047144	000400	000400
12121	047146	001000	001000
12122	047150	002000	002000
12123	047152	004000	004000
12124	047154	010000	010000
12125	047156	020000	020000
12126	047160	040000	040000
12127	047162	100000	100000
12128			
12129			
12130			
12131	047164		
12132	047164	177776	177776
12133	047166	177775	177775
12134	047170	177773	177773
12135	047172	177767	177767
12136	047174	177757	177757
12137	047176	177737	177737
12138	047200	177677	177677
12139	047202	177577	177577
12140	047204	177377	177377
12141	047206	176777	176777
12142	047210	175777	175777
12143	047212	173777	173777
12144	047214	167777	167777
12145	047216	157777	157777
12146	047220	137777	137777
12147	047222	077777	077777
12148			
12149			

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

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

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

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

;DATA PATTERN 13

F02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 224
DATA PATTERNS

SEQ 0224

12150			PAT13: ALTERNATING 0-1
12151	047224		052525
12152	047224	052525	052525
12153	047226	052525	052525
12154	047230	052525	052525
12155	047232	052525	052525
12156	047234	052525	052525
12157	047236	052525	052525
12158	047240	052525	052525
12159	047242	052525	052525
12160	047244	052525	052525
12161	047246	052525	052525
12162	047250	052525	052525
12163	047252	052525	052525
12164	047254	052525	052525
12165	047256	052525	052525
12166	047260	052525	052525
12167	047262	052525	052525

12168			
12169			
12170			; DATA PATTERN 14
12171	047264		PAT14: ALTERNATING 1-0
12172	047264	125252	125252
12173	047266	125252	125252
12174	047270	125252	125252
12175	047272	125252	125252
12176	047274	125252	125252
12177	047276	125252	125252
12178	047300	125252	125252
12179	047302	125252	125252
12180	047304	125252	125252
12181	047306	125252	125252
12182	047310	125252	125252
12183	047312	125252	125252
12184	047314	125252	125252
12185	047316	125252	125252
12186	047320	125252	125252
12187	047322	125252	125252

12188			
12189			
12190			; DATA PATTERN 15
12191	047324		PAT15: SHIFTING ZEROS AND ONES
12192	047324	000001	000001
12193	047326	000003	000003
12194	047330	000007	000007
12195	047332	000017	000017
12196	047334	000037	000037
12197	047336	000077	000077
12198	047340	000177	000177
12199	047342	000377	000377
12200	047344	000777	000777
12201	047346	001777	001777
12202	047350	003777	003777
12203	047352	007777	007777
12204	047354	017777	017777
12205	047356	037777	037777

G02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 225
DATA PATTERNS

SEQ 0225

12206	047360	077777	077777
12207	047362	177777	177777
12208			
12209			
12210			; DATA PATTERN 16
12211	047364		COMPOSITE ROTATING
12212	047364	072307	PAT16: 072307
12213	047366	135143	135143
12214	047370	156461	156461
12215	047372	167230	167230
12216	047374	073514	073514
12217	047376	035646	035646
12218	047400	016723	016723
12219	047402	107351	107351
12220	047404	143564	143564
12221	047406	061672	061672
12222	047410	030735	030735
12223	047412	114356	114356
12224	047414	046167	046167
12225	047416	123073	123073
12226	047420	151453	151453
12227	047422	164616	164616
12228			

H02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 226
FIELDS AND VARIABLES FOR OPERATION CHECKING

SEQ 0226

12229		.SBTTL FIELDS AND VARIABLES FOR OPERATION CHECKING	
12230	024000	CS1ERBIT=24000	; CS1 ERROR BITS SPAR & CTO
12231	177400	CS2ERBIT=177400	; CS2 ERROR BITS
12232			; DLT, WCE, UPE, NED, NEM
12233			; PGE, MOS, UFE
12234	000070	DSERBIT=70	; DS ERROR BITS
12235			; SPDLS, DROT, ACLO
12236			
12237	047424 000000	EXPWC: .WORD 0	; EXPECTED WORD COUNT (GIVEN)
12238	047426 000000	EXPUBA: .WORD 0	; EXPECTED UPPER BA (COMPUTED)
12239	047430 000000	EXPBA: .WORD 0	; EXPECTED BUS ADDRESS (COMPUTED)
12240	047432 000000	EXPCYL: .WORD 0	; EXPECTED CYLINDER (COMPUTED)
12241	047434 000000	EXPSEC: .WORD 0	; EXPECTED SECTOR (COMPUTED)
12242	047436 000000	EXPTRK: .WORD 0	; EXPECTED TRACK (COMPUTED)
12243			
12244	047440 000000	REALWC: .WORD 0	; WORD COUNT AT END OF OPERATION
12245	047442 000000	REALUB: .WORD 0	; REAL UPPER BA
12246	047444 000000	REALBA: .WORD 0	; BUS ADDRESS
12247	047446 000000	REALCY: .WORD 0	; CYLINDER
12248	047450 000000	REALTRK: .WORD 0	; TRACK
12249	047452 000000	REALSEC: .WORD 0	; SECTOR
12250			
12251	047454 000000	GRP1ER: .WORD 0	; GROUP 1 ERROR FIELDS
12252	000001	SPARERR=BIT0	; CONTROLLER DETECTED DRIVE BUS PARITY ERR
12253	000002	SKIERR=BIT1	; SEEK INCOMPLETE
12254	000004	NXFERR=BIT2	; NON-EXECUTABLE DRIVE FUNCTION
12255	000010	DRPARERR=BIT3	; DRIVE DETECTED DRIVE BUS PARITY ERROR
12256	000020	FMTERR=BIT4	; FORMAT ERROR
12257	000040	DTYERR=BIT5	; DRIVE TYPE ERROR
12258	000100	ACLOERR=BIT6	; AC LOW ERROR
12259	000200	SPDERR=BIT7	; SPEED LOSS ERROR
12260	000400	DROTERR=BIT8	; DRIVE OFF TRACK ERROR
12261	001000	COERR=BIT9	; CYLINDER OVER FLOW ERROR
12262	002000	IDAERR=BIT10	; ILLEGAL DISK ADDRESS ERROR
12263	004000	WLERR=BIT11	; WRITE LOCK ERROR
12264	010000	DTERR=BIT12	; DRIVE TIMING ERROR
12265	020000	NCERWE=BIT13	; NO CERR WITH ERROR SET ERROR
12266	040000	UNSEERR=BIT14	; DRIVE UNSAFE ERROR
12267	100000	CERNER=BIT15	; CERR BUT NO ERROR SET ERROR
12268			
12269	047456 000000	GRP2ER: .WORD 0	; GROUP 2 ERROR FIELD
12270	000001	ECHERR=BIT0	; ECC HARD ERROR
12271	000002	DCKERR=BIT1	; DATA CHECK ERROR
12272	000004	WCKERR=BIT2	; WRITE CHECK ERROR
12273	000010	DLTERR=BIT3	; DATA LATE ERROR
12274	000020	OPIERR=BIT4	; OPERATION INCOMPLETE ERROR
12275	000040	HVRCERR=BIT5	; HEADER VRC ERROR
12276	000100	BSERR=BIT6	; BAD SECTOR ERROR
12277			
12278	047460 000000	GRP3ER: .WORD 0	; GROUP 3 ERROR FIELD
12279	000001	NEDERR=BIT0	; NON-EXISTANT DRIVE ERROR
12280	000002	CTOERR=BIT1	; CONTROLLER TIME OUT ERROR
12281	000004	UFERR=BIT2	; UNIT FIELD ERROR
12282	000010	MDSERR=BIT3	; MULTIPLE DRIVE SELECT ERROR
12283	000020	PGERR=BIT4	; PROGRAMMING ERROR
12284	000040	NEMERR=BIT5	; NON-EXISTANT MEMORY ERROR

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR&KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 227
FIELDS AND VARIABLES FOR OPERATION CHECKING

SEQ 0227

12285		000100	UPERR= BIT6	;UNIBUS PARITY ERROR
12286		000200	ILFERR= BIT7	;ILLEGAL FUNCTION ERROR.
12287				
12288	047462	000000	GRP4ER: .WORD 0	;GROUP 4 ERROR FIELD
12289		000001	WCERR= BIT0	;WORD COUNT ERROR FLAG
12290		000002	UBAERR= BIT1	;UPPER BA ERROR
12291		000004	BAERR= BIT2	;BUS ADDRESS ERROR FLAG
12292		000010	CYLERR= BIT3	;CYL ADDRESS ERROR FLAG
12293		000020	TRKERR= BIT4	;TRACK ADDRESS ERROR FLAG
12294		000040	SECERR= BIT5	;SECTOR ADDRESS ERROR FLAG
12295				
12296	047464	053774	GRP4MS: .WORD EM10	
12297	047466	054047	.WORD EM11A	
12298	047470	054021	.WORD EM11	
12299	047472	054123	.WORD EM12	
12300	047474	054156	.WORD EM13	
12301	047476	054206	.WORD EM14	
12302				
12303	047500	054240	GRP3MS: .WORD EM15	
12304	047502	054263	.WORD EM16	
12305	047504	054306	.WORD EM17	
12306	047506	054327	.WORD EM18	
12307	047510	054354	.WORD EM19	
12308	047512	054376	.WORD EM20	
12309	047514	054422	.WORD EM21	
12310	047516	054446	.WORD EM22	
12311				
12312	047520	054467	GRP2MS: .WORD EM23	
12313	047522	054500	.WORD EM24	
12314	047524	054513	.WORD EM25	
12315	047526	054527	.WORD EM26	
12316	047530	054541	.WORD EM27	
12317	047532	054566	.WORD EM28	
12318	047534	054601	.WORD EM29	
12319				
12320	047536	054622	GRP1MS: .WORD EM30	
12321	047540	054675	.WORD EM31	
12322	047542	054715	.WORD EM32	
12323	047544	054753	.WORD EM33	
12324	047546	055021	.WORD EM34	
12325	047550	055036	.WORD EM35	
12326	047552	055057	.WORD EM36	
12327	047554	055066	.WORD EM37	
12328	047556	055111	.WORD EM38	
12329	047560	055131	.WORD EM39	
12330	047562	055153	.WORD EM40	
12331	047564	055205	.WORD EM41	
12332	047566	055226	.WORD EM42	
12333	047570	055251	.WORD EM43	
12334	047572	055313	.WORD EM44	
12335	047574	055330	.WORD EM45	
12336				
12337	047576	000000	GPSUMF: .WORD 0	;GROUP ERROR SUMMARY FLAGS
12338		000001	GRP1ST= BIT0	;GROUP 1 ERROR SET
12339		000002	GRP2ST= BIT1	;GROUP 2 ERROR SET
12340		000004	GRP3ST= BIT2	;GROUP 3 ERROR SET

J02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 228
FIELDS AND VARIABLES FOR OPERATION CHECKING

SEQ 0228

12341	000010	GP1NR=	BIT3	;GROUP 1 ERROR NOT RECEIVED
12342	000020	GP2NR=	BIT4	;GROUP 2 ERROR NOT RECEIVED
12343	000040	GP3NR=	BIT5	;GROUP 3 ERROR NOT RECEIVED
12344	040000	DRSTER=	BIT14	;ERROR IN GETTING DRIVE STATUS FLAG.
12345	100000	REPNR=	BIT15	;REPORTING NOT RECEIVED SWITCH
12346				
12347		.SBTTL	TABLE OF OPERATION MESSAGE ADDRESS	
12348		;*	THIS TABLE CONTAINS THE ADDRESS OF ASCIZ FIELDS THAT ARE	
12349		;*	USED IN REPORTING TO IDENTIFY THE OPERATION BEING PERFORMED.	
12350				
12351	047600	C 7640	CMNDLB: .WORD	OPER00 ;ADDRESS OF SELECT MESSAGE
12352	047602	047655	.WORD	OPER02 ;PACK ACK
12353	047604	047666	.WORD	OPER04 ;DRIVE CLEAR
12354	047606	047702	.WORD	OPER06 ;UNLOAD
12355	047610	047711	.WORD	OPER10 ;START SPINDLE
12356	047612	047727	.WORD	OPER12 ;RECALIBRATE
12357	047614	047743	.WORD	OPER14 ;OFFSET
12358	047616	047752	.WORD	OPER16 ;SEEK
12359	047620	047757	.WORD	OPER20 ;READ DATA
12360	047622	047771	.WORD	OPER22 ;WRITE DATA
12361	047624	050004	.WORD	OPER24 ;READ HEADER
12362	047626	050021	.WORD	OPER26 ;WRITE HEADER
12363	047630	050037	.WORD	OPER30 ;WRITE CHECK
12364	047632	050053	.WORD	OPER32 ;ILLEGAL OPERATION 33
12365	047634	050077	.WORD	OPER34 ;35
12366	047636	050123	.WORD	OPER36 ;37
12367				
12368			.SBTTL	OPERATION MESSAGES
12369	047640	051104	053111	020105
12370	047646	042523	042514	052103
12371	047654	000		
12372	047655	120	041501	020113
12373	047662	041501	000113	
12374	047666	051104	053111	020105
12375	047674	046103	040505	000122
12376	047702	047125	047514	042101
12377	047710	000		
12378	047711	123	040524	052122
12379	047716	051440	044520	042116
12380	047724	042514	000	
12381	047727	122	041505	046101
12382	047734	041111	040522	042524
12383	047742	000		
12384	047743	117	043106	042523
12385	047750	000124		
12386	047752	042523	045505	000
12387	047757	122	040505	020104
12388	047764	040504	040524	000
12389	047771	127	044522	042524
12390	047776	042040	052101	000101
12391	050004	042522	042101	044040
12392	050012	040505	042504	051522
12393	050020	000		
12394	050021	127	044522	042524
12395	050026	044040	040505	042504
12396	050034	051522	000	

K02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 229
OPERATION MESSAGES

SEQ 0229

12397	050037	127	044522	042524	OPER30: .ASCIZ /WRITE CHECK/
12398	050044	041440	042510	045503	
12399	050052	000			
12400	050053	111	046114	043505	OPER32: .ASCIZ /ILLEGAL FUNCTION 33/
12401	050060	046101	043040	047125	
12402	050066	052103	047511	020116	
12403	050074	031463	000		
12404	050077	111	046114	043505	OPER34: .ASCIZ /ILLEGAL FUNCTION 35/
12405	050104	046101	043040	047125	
12406	050112	052103	047511	020116	
12407	050120	032463	000		
12408	050123	111	046114	043505	OPER36: .ASCIZ /ILLEGAL FUNCTION 37/
12409	050130	046101	043040	047125	
12410	050136	052103	047511	020116	
12411	050144	033463	000		
12412	050147	127	044522	042524	OPER37: .ASCIZ /WRITE DATA ABORTED WITH BSE/
12413	050154	042040	052101	020101	
12414	050162	041101	051117	042524	
12415	050170	020104	044527	044124	
12416	050176	041040	042523	000	
12417	050203				OPER40:
12418	050203	127	044522	042524	OPER41: .ASCIZ /WRITE CHECK ABORTED WITH WCE/
12419	050210	041440	042510	045503	
12420	050216	040440	047502	052122	
12421	050224	042105	053440	052111	
12422	050232	020110	041527	000105	
12423	050240	051127	052111	020105	OPER42: .ASCIZ /WRITE DATA ABORTED WITH NON-EXISTANT MEMORY/
12424	050246	040504	040524	040440	
12425	050254	047502	052122	042105	
12426	050262	053440	052111	020110	
12427	050270	047516	026516	054105	
12428	050276	051511	040524	052116	
12429	050304	046440	046505	051117	
12430	050312	000131			
12431	050314	042522	042101	042040	OPER43: .ASCIZ /READ DATA ABORTED WITH NON-EXISTANT MEMORY/
12432	050322	052101	020101	041101	
12433	050330	051117	042524	020104	
12434	050336	044527	044124	047040	
12435	050344	047117	042455	044530	
12436	050352	052123	047101	020124	
12437	050360	042515	047515	054522	
12438	050366	000			
12439	050367	127	044522	042524	OPER44: .ASCIZ /WRITE DATA ABORTED WITH UNIBUS PARITY ERROR/
12440	050374	042040	052101	020101	
12441	050402	041101	051117	042524	
12442	050410	020104	044527	044124	
12443	050416	052440	044516	052502	
12444	050424	020123	040520	044522	
12445	050432	054524	042440	051122	
12446	050440	051117	000		
12447					

L02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 230
ASCII MESSAGES

SEQ 0230

12448					.SBTTL ASCII MESSAGES
12449					
12450	050443	040	000040		SPACE2: .ASCIZ / /
12451	050446	005015	045522	030466	OPR001: .ASCIZ <15><12>/RK611 BUS ADDRESS (/
12452	050454	020061	052502	020123	
12453	050462	042101	051104	051505	
12454	050470	020123	020050	000	
12455	050475	040	020051	020075	OPR002: .ASCIZ /) = /
12456	050502	000			
12457	050503	122	033113	030461	OPR003: .ASCIZ /RK611 VECTOR ADDRESS (/
12458	050510	053040	041505	047524	
12459	050516	020122	042101	051104	
12460	050524	051505	020123	020050	
12461	050532	000			
12462	050533	122	033113	030461	OPR004: .ASCIZ /RK611 PRIORITY (/
12463	050540	050040	044522	051117	
12464	050546	052111	020131	020050	
12465	050554	000			
12466	050555	111	051516	043125	OPR005: .ASCIZ /INSUFFICIENT MEMORY. PROGRAM ABORTING./<15><12>
12467	050562	044506	044503	047105	
12468	050570	020124	042515	047515	
12469	050576	054522	020056	051120	
12470	050604	043517	040522	020115	
12471	050612	041101	051117	044524	
12472	050620	043516	006456	000012	
12473	050626	005015	047524	041040	OPR006: .ASCII <15><12>/TO BYPASS TESTING DRIVE 0, PLACE IT OFF-LINE/
12474	050634	050131	051501	020123	
12475	050642	042524	052123	047111	
12476	050650	020107	051104	053111	
12477	050656	020105	026060	050040	
12478	050664	040514	042503	044440	
12479	050672	020124	043117	026506	
12480	050700	044514	042516		
12481	050704	005015	047524	052040	.ASCIZ <15><12>/TO TEST DRIVE 0, REPLACE PROGRAM PACK WITH SCRATCH PACK/
12482	050712	051505	020124	051104	
12483	050720	053111	020105	026060	
12484	050726	051040	050105	040514	
12485	050734	042503	050040	047522	
12486	050742	051107	046501	050040	
12487	050750	041501	020113	044527	
12488	050756	044124	051440	051103	
12489	050764	052101	044103	050040	
12490	050772	041501	000113		
12491	050776	005015	051104	053111	OPR007: .ASCII <15><12>/DRIVE 0 WILL NOT BE TESTED. TO TEST DRIVE 0,/<15><12>
12492	051004	020105	020060	044527	
12493	051012	046114	047040	052117	
12494	051020	041040	020105	042524	
12495	051026	052123	042105	020056	
12496	051034	047524	052040	051505	
12497	051042	020124	051104	053111	
12498	051050	020105	026060	005015	
12499	051056	042522	052123	051101	.ASCIZ /RESTART AT 214 AND MOUNT SCRATCH PACK AS DIRECTED./
12500	051064	020124	052101	031040	
12501	051072	032061	040440	042116	
12502	051100	046440	052517	052116	
12503	051106	051440	051101	052101	

M02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 231
ASCII MESSAGES

SEQ 0231

12504	051114	044103	050040	041501
12505	051122	020113	051501	042040
12506	051130	051111	041505	042524
12507	051136	027104	000	
12508	051141	015	047012	020117
12509	051146	051104	053111	051505
12510	051154	040440	040526	046111
12511	051162	041101	042514	043040
12512	051170	051117	052040	051505
12513	051176	044524	043516	020056
12514	051204	051120	043517	040522
12515	051212	020115	041101	051117
12516	051220	042524	000104	
12517	051224	005015	044124	020105
12518	051232	047506	046114	053517
12519	051240	047111	030107	051104
12520	051246	053111	051505	053440
12521	051254	046111	020114	042502
12522	051262	052040	051505	042524
12523	051270	006504	000012	
12524	051274	005015	047516	050040
12525	051302	051101	052111	020131
12526	051310	050117	044524	047117
12527	051316	043040	051117	046440
12528	051324	046505	051117	020131
12529	051332	051101	040505	044440
12530	051340	020116	051525	006505
12531	051346	012		
12532	051347	106	051117	052040
12533	051354	051505	020124	020055
12534	051362	047125	041111	051525
12535	051370	050040	051101	052111
12536	051376	020131	051105	047522
12537	051404	020122	047101	006504
12538	051412	012		
12539	051413	062	024064	024470
12540	051420	051440	041505	047524
12541	051426	020122	047506	046522
12542	051434	052101	042040	052101
12543	051442	020101	043130	051105
12544	051450	052040	051505	020124
12545	051456	054502	040520	051523
12546	051464	042105	005015	000
12547	051471	015	046412	046505
12548	051476	051117	020131	044523
12549	051504	042532	047040	052117
12550	051512	046040	051101	042507
12551	051520	042440	047516	043525
12552	051526	020110	047506	020122
12553	051534	052502	020123	042101
12554	051542	051104	051505	020123
12555	051550	044502	051524	030440
12556	051556	020066	020046	033461
12557	051564	052040	051505	051524
12558	051572	005015		
12559	051574	046101	020114	040504

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

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

OPR010: .ASCII <15><12>/NO PARITY OPTION FOR MEMORY AREA IN USE/<15><12>

.ASCII /FOR TEST - 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 >/

N02

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 15:34 PAGE 232
ASCII MESSAGES

SEQ 0232

12560	051602	040524	054040	042536
12561	051610	020122	042524	052123
12562	051616	020123	044527	044124
12563	051624	040440	042104	020122
12564	051632	000076		
12565	051634	031063	000113	
12566	051640	032066	000113	
12567	051644	033071	000113	
12568	051650	041040	050131	051501
12569	051656	042523	006504	000012
12570	051664	005015	020012	020040
12571	051672	020040	020040	025052
12572	051700	020052	040503	052125
12573	051706	047511	020116	025052
12574	051714	006452	005012	
12575	051720	044124	051511	050040
12576	051726	047522	051107	046501
12577	051734	051440	047510	046125
12578	051742	020104	042502	044040
12579	051750	046101	042524	020104
12580	051756	054502	052040	050131
12581	051764	047111	020107	041536
12582	051772	006456	012	
12583	051775	111	020106	040510
12584	052002	052114	042105	052440
12585	052010	044523	043516	052040
12586	052016	042510	044040	046101
12587	052024	020124	042513	026131
12588	052032	052040	042510	051440
12589	052040	040524	042524	047440
12590	052046	020106	044124	020105
12591	052054	051104	053111	006505
12592	052062	012		
12593	052063	117	020122	040503
12594	052070	052122	044522	043504
12595	052076	020105	040503	047116
12596	052104	052117	041040	020105
12597	052112	051120	042105	041511
12598	052120	042524	027104	005015
12599	052126	012		
12600	052127	101	046114	042040
12601	052134	044522	042526	020123
12602	052142	047524	041040	020105
12603	052150	042524	052123	042105
12604	052156	046440	051525	020124
12605	052164	042502	047440	026516
12606	052172	044514	042516	006454
12607	052200	012		
12608	052201	122	040505	054504
12609	052206	020054	047101	020104
12610	052214	051127	052111	021105
12611	052222	047514	045503	051040
12612	052230	051505	052105	006456
12613	052236	012		
12614	052237	101	054516	042040
12615	052244	044522	042526	047040

OPR012: .ASCII /32K/
OPR013: .ASCII /64K/
OPR014: .ASCII /96K/
OPR015: .ASCII / BYPASSED/(15)(12)

OPR016: .ASCII (15)(12)(12)/ *** CAUTION ***/(15)(12)(12)

.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)

B03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 233
ASCII MESSAGES

SEQ 0233

12616	052252	052117	052040	020117
12617	052260	042502	052040	051505
12618	052266	042524	020104	052515
12619	052274	052123	041040	020105
12620	052302	043117	026506	044514
12621	052310	042516	006456	005012
12622	052316	047516	042524	020072
12623	052324	047062	020104	047101
12624	052332	020104	052523	051502
12625	052340	050505	042525	052116
12626	052346	050040	051501	020123
12627	052354	052522	020116	044524
12628	052362	042515	044440	006523
12629	052370	012		
12630	052371	040	020040	020040
12631	052376	040440	050120	047522
12632	052404	020130	020062	044515
12633	052412	020116	030063	051440
12634	052420	041505	043040	051117
12635	052426	042440	041501	020110
12636	052434	051104	053111	027105
12637	052442	005015	000	
12638	052445	015	043012	051111
12639	052452	052123	031040	033065
12640	052460	051440	041505	047524
12641	052466	051522	047040	052117
12642	052474	041040	042523	042440
12643	052502	051122	051117	043040
12644	052510	042522	027105	
12645	052514	040515	044530	052515
12646	052522	020115	040504	040524
12647	052530	052040	040522	051516
12648	052536	042506	020122	042524
12649	052544	052123	041040	050131
12650	052552	051501	042523	006504
12651	052560	000012		
12652	052562	020040	020040	006440
12653	052570	047412	046116	020131
12654	052576	020061	051104	053111
12655	052604	027105	047440	042526
12656	052612	046122	050101	042520
12657	052620	020104	050117	051105
12658	052626	052101	047511	051516
12659	052634	041040	050131	051501
12660	052642	042523	006504	000012
12661	052650	005015	041523	050117
12662	052656	035105	041440	030510
12663	052664	024040	051124	043511
12664	052672	026051	042440	031465
12665	052700	034055	020073	044103
12666	052706	026062	042440	034464
12667	052714	031055	024040	041501
12668	052722	041440	052517	046120
12669	052730	026105	027040	053062
12670	052736	041457	024515	
12671	052742	005015	042101	052512

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

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

OPR017: .ASCII <15><12>/FIRST 256 SECTORS NOT BSE ERROR FREE./

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

OPR018: .ASCIZ / /<15><12>/ONLY 1 DRIVE. OVERLAPPED OPERATIONS BYPASSED/<15><12>

OPR019: .ASCII <15><12>2SCOPE: CH1 (TRIG), E53-8; CH2, E49-2 (AC COUPLE, .2V/CM)2

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

C03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 234
ASCII MESSAGES

SEQ 0234

12672	052750	052123	051040	031067
12673	052756	043040	051117	041440
12674	052764	047117	052123	047101
12675	052772	020124	042514	042526
12676	053000	020114	047117	041440
12677	053006	031110	005015	000
12678	053013	015	050012	047522
12679	053020	051107	046501	044040
12680	053026	046101	020124	042520
12681	053034	042116	047111	020107
12682	053042	020055	040503	052122
12683	053050	044522	043504	020105
12684	053056	047506	046522	052101
12685	053064	041040	044505	043516
12686	053072	041440	051117	042522
12687	053107	052103	042105	005015
12688	053106	000		
12689	053107	015	025012	025052
12690	053114	025052	020040	051120
12691	053122	043517	040522	020115
12692	053130	040510	052114	042105
12693	053136	020040	025052	025052
12694	053144	006452	000012	
12695	053150	040503	052122	044522
12696	053156	043504	020105	047506
12697	053164	046522	052101	041440
12698	053172	051117	042522	052103
12699	053200	047511	020116	040506
12700	053206	046111	042105	005015
12701	053214	000		
12702	053215	015	050012	047522
12703	053222	051107	046501	040440
12704	053230	047502	052122	047111
12705	053236	020107	042502	040503
12706	053244	051525	020105	051105
12707	053252	047522	020122	044124
12708	053260	042522	044123	046117
12709	053266	020104	054105	042503
12710	053274	042105	042105	005015
12711	053302	000		
12712				
12713				
12714	053303	106	052101	046101
12715	053310	047055	047117	042440
12716	053316	044530	052123	047101
12717	053324	020124	042515	047515
12718	053332	054522	040440	020124
12719	053340	045522	030466	020061
12720	053346	040502	042523	040440
12721	053354	042104	042522	051523
12722	053362	000		
12723	053363	106	052101	046101
12724	053370	053455	044522	042524
12725	053376	051040	040505	054504
12726	053404	040440	042116	044440
12727	053412	020105	044504	020104

OPR020: .ASCIZ <15><12>/PROGRAM HALT PENDING - CARTRIDGE FORMAT BEING CORRECTED/<15><12>

OPR021: .ASCIZ <15><12>/***** PROGRAM HALTED *****/<15><12>

OPR022: .ASCIZ /CARTRIDGE FORMAT CORRECTION FAILED/<15><12>

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

D03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 235
ERROR MESSAGES

SEQ 0235

12728	053420	047516	020124	040503	
12729	053426	051525	020105	047111	
12730	053434	042524	051122	050125	
12731	053442	000124			
12732	053444	040506	040524	026514	EM3: .ASCIZ /FATAL-PARITY ERROR TRAP. PC AT ERROR = /
12733	053452	040520	044522	054524	
12734	053460	042440	051122	051117	
12735	053466	052040	040522	027120	
12736	053474	050040	020103	052101	
12737	053502	042440	051122	051117	
12738	053510	036440	000040		
12739	053514	054105	042520	052103	EM4: .ASCIZ /EXPECTED INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS:/
12740	053522	042105	044440	052116	
12741	053530	051105	052522	052120	
12742	053536	042040	042111	047040	
12743	053544	052117	047440	041503	
12744	053552	051125	047440	020122	
12745	053560	040527	020123	040514	
12746	053566	042524	020056	047503	
12747	053574	046515	047101	020104	
12748	053602	040527	035123	000	
12749	053607	123	041125	054523	EM5: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET ERROR/
12750	053614	052123	046505	041440	
12751	053622	042514	051101	042040	
12752	053630	042111	047040	052117	
12753	053636	051040	051505	052105	
12754	053644	042440	051122	051117	
12755	053652	000			
12756	053653	123	041125	054523	EM6: .ASCIZ /SUBSYSTEM CLEAR DID NOT RESET DEVICE INTERRUPT/
12757	053660	052123	046505	041440	
12758	053666	042514	051101	042040	
12759	053674	042111	047040	052117	
12760	053702	051040	051505	052105	
12761	053710	042040	053105	041511	
12762	053716	020105	047111	042524	
12763	053724	051122	050125	000124	
12764	053732	052523	051502	051531	EM7: .ASCIZ /SUBSYSTEM CLEAR DID NOT SET READY/
12765	053740	042524	020115	046103	
12766	053746	040505	020122	044504	
12767	053754	020104	047516	020124	
12768	053762	042523	020124	042522	
12769	053770	042101	000131		
12770	053774	047527	042122	041440	EM10: .ASCIZ /WORD COUNT INCORRECT/
12771	054002	052517	052116	044440	
12772	054010	041516	051117	042522	
12773	054016	052103	000		
12774	054021	102	051525	040440	EM11: .ASCIZ /BUS ADDRESS INCORRECT/
12775	054026	042104	042522	051523	
12776	054034	044440	041516	051117	
12777	054042	042522	052103	000	
12778	054047	125	050120	051105	EM11A: .ASCIZ /UPPER BUS ADDRESS BITS INCORRECT (BA16, 17)/
12779	054054	041040	051525	040440	
12780	054062	042104	042522	051523	
12781	054070	041040	052111	020123	
12782	054076	047111	047503	051122	
12783	054104	041505	020124	041050	

E03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 236
ERROR MESSAGES

SEQ 0236

12784	054112	030501	026066	030440		
12785	054120	024467	000			
12786	054123	103	046131	047111	EM12:	.ASCIZ /CYLINDER ADDRESS INCORRECT/
12787	054130	042504	020122	042101		
12788	054136	051104	051505	020123		
12789	054144	047111	047503	051122		
12790	054152	041505	000124			
12791	054156	051124	041501	020113	EM13:	.ASCIZ /TRACK ADDRESS INCORRECT/
12792	054164	042101	051104	051505		
12793	054172	020123	047111	047503		
12794	054200	051122	041505	000124		
12795	0542	042523	052103	051117	EM14:	.ASCIZ /SECTOR ADDRESS INCORRECT./
12796	05421	040440	042104	042522		
12797	05422	051523	044440	041516		
12798	054230	051117	042522	052103		
12799	054236	000056				
12800	054240	047516	026516	054105	EM15:	.ASCIZ /NON-EXISTANT DRIVE/
12801	054246	051511	040524	052116		
12802	054254	042040	044522	042526		
12803	054262	000				
12804	054263	103	047117	051124	EM16:	.ASCIZ /CONTROLLER TIMEOUT/
12805	054270	046117	042514	020122		
12806	054276	044524	042515	052517		
12807	054304	000124				
12808	054306	047125	052111	043040	EM17:	.ASCIZ /UNIT FIELD ERROR/
12809	054314	042511	042114	042440		
12810	054322	051122	051117	000		
12811	054327	115	046125	050111	EM18:	.ASCIZ /MULIPLE DRIVE SELECT/
12812	054334	042514	042040	044522		
12813	054342	042526	051440	046105		
12814	054350	041505	000124			
12815	054354	051120	043517	040522	EM19:	.ASCIZ /PROGRAMMING ERROR/
12816	054362	046515	047111	020107		
12817	054370	051105	047522	000122		
12818	054376	047516	026516	054105	EM20:	.ASCIZ /NON-EXISTANT MEMORY/
12819	054404	051511	040524	052116		
12820	054412	046440	046505	051117		
12821	054420	000131				
12822	054422	047125	041111	051525	EM21:	.ASCIZ /UNIBUS PARITY ERROR/
12823	054430	050040	051101	052111		
12824	054436	020131	051105	047522		
12825	054444	000122				
12826	054446	046111	042514	040507	EM22:	.ASCIZ /ILLEGAL FUNCTION/
12827	054454	020114	052506	041516		
12828	054462	044524	047117	000		
12829	054467	105	041503	044040	EM23:	.ASCIZ /ECC HARD/
12830	054474	051101	000104			
12831	054500	040504	040524	041440	EM24:	.ASCIZ /DATA CHECK/
12832	054506	042510	045503	000		
12833	054513	127	044522	042524	EM25:	.ASCIZ /WRITE CHECK/
12834	054520	041440	042510	045503		
12835	054526	000				
12836	054527	104	052101	020101	EM26:	.ASCIZ /DATA LATE/
12837	054534	040514	042524	000		
12838	054541	117	042520	040522	EM27:	.ASCIZ /OPERATION INCOMPLETE/
12839	054546	044524	047117	044440		

F03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006)
ERROR MESSAGES

31-JAN-77 16:34 PAGE 237

SEQ 0237

12840	054554	041516	046517	046120	
12841	054562	052105	000105		
12842	054566	042510	042101	051105	EM28: .ASCIZ /HEADER VRC/
12843	054574	053040	041522	000	
12844	054601	102	042101	051440	EM29: .ASCIZ /BAD SECTOR ERROR/
12845	054606	041505	047524	020122	
12846	054614	051105	047522	000122	
12847	054622	047503	052116	047522	EM30: .ASCIZ /CONTROLLER DETECTED DRIVE BUS PARITY ERROR/
12848	054630	046114	051105	042040	
12849	054636	052105	041505	042524	
12850	054644	020104	051104	053111	
12851	054652	020105	052502	020123	
12852	054660	040520	044522	054524	
12853	054666	042440	051122	051117	
12854	054674	000			
12855	054675	123	042505	020113	EM31: .ASCIZ /SEEK INCOMPLETE/
12856	054702	047111	047503	050115	
12857	054710	042514	042524	000	
12858	054715	116	047117	042455	EM32: .ASCIZ /NON-EXECUTABLE DRIVE FUNCTION/
12859	054722	042530	052503	040524	
12860	054730	046102	020105	051104	
12861	054736	053111	020105	052506	
12862	054744	041516	044524	047117	
12863	054752	000			
12864	054753	104	044522	042526	EM33: .ASCIZ /DRIVE DETECTED DRIVE BUS PARITY ERROR/
12865	054760	042040	052105	041505	
12866	054766	042524	020104	051104	
12867	054774	053111	020105	052502	
12868	055002	020123	040520	044522	
12869	055010	054524	042440	051122	
12870	055016	051117	000		
12871	055021	106	051117	040515	EM34: .ASCIZ /FORMAT ERROR/
12872	055026	020124	051105	047522	
12873	055034	000122			
12874	055036	051104	053111	020105	EM35: .ASCIZ /DRIVE TYPE ERROR/
12875	055044	054524	042520	042440	
12876	055052	051122	051117	000	
12877	055057	101	020103	047514	EM36: .ASCIZ /AC LOW/
12878	055064	000127			
12879	055066	050123	047111	046104	EM37: .ASCIZ /SPINDLE SPEED LOSS/
12880	055074	020105	050123	042505	
12881	055102	020104	047514	051523	
12882	055110	000			
12883	055111	104	044522	042526	EM38: .ASCIZ /DRIVE OFF TRACK/
12884	055116	047440	043106	052040	
12885	055124	040522	045503	000	
12886	055131	103	046131	047111	EM39: .ASCIZ /CYLINDER OVERFLOW/
12887	055136	042504	020122	053117	
12888	055144	051105	046106	053517	
12889	055152	000			
12890	055153	111	046114	043505	EM40: .ASCIZ /ILLEGAL DISK PACK ADDRESS/
12891	055160	046101	042040	051511	
12892	055166	020113	040520	045503	
12893	055174	040440	042104	042522	
12894	055202	051523	000		
12895	055205	127	044522	042524	EM41: .ASCIZ /WRITE LOCK ERROR/

G03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 238
ERROR MESSAGES

SEQ 0238

12896	055212	046040	041517	020113	
12897	055220	051105	047522	000122	
12898	055226	051104	053111	020105	EM42: .ASCIZ /DRIVE TIMING ERROR/
12899	055234	044524	044515	043516	
12900	055242	042440	051122	051117	
12901	055250	000			
12902	055251	116	020117	042503	EM43: .ASCIZ /NO CERR WITH SOME OTHER ERROR SET/
12903	055256	051122	053440	052111	
12904	055264	020110	047523	042515	
12905	055272	047440	044124	051105	
12906	055300	042440	051122	051117	
12907	055306	051440	052105	000	
12908	055313	104	044522	042526	EM44: .ASCIZ /DRIVE UNSAFE/
12909	055320	052440	051516	043101	
12910	055326	000105			
12911	055330	042503	051122	051440	EM45: .ASCIZ /CERR SET BUT NO OTHER ERROR SET/
12912	055336	052105	041040	052125	
12913	055344	047040	020117	052117	
12914	055352	042510	020122	051105	
12915	055360	047522	020122	042523	
12916	055366	000124			
12917					
12918	055370	053126	042040	042111	EM46: .ASCIZ /VV DID NOT SET WITH PACK ACK/
12919	055376	047040	052117	051440	
12920	055404	052105	053440	052111	
12921	055412	020110	040520	045503	
12922	055420	040440	045503	000	
12923	055425	123	040524	052524	EM47: .ASCIZ /STATUS VALID SET ON SELECT TO NON-EXISTANT DRIVE/
12924	055432	020123	040526	044514	
12925	055440	020104	042523	020124	
12926	055446	047117	051440	046105	
12927	055454	041505	020124	047524	
12928	055462	047040	047117	042455	
12929	055470	044530	052123	047101	
12930	055476	020124	051104	053111	
12931	055504	000105			
12932	055506	052123	052101	051525	EM48: .ASCIZ /STATUS VALID RESET ON SELECT TO EXISTANT DRIVE/
12933	055514	053040	046101	042111	
12934	055522	051040	051505	052105	
12935	055530	047440	020116	042523	
12936	055536	042514	052103	052040	
12937	055544	020117	054105	051511	
12938	055552	040524	052116	042040	
12939	055560	044522	042526	000	
12940	055565	123	040524	052524	EM49: .ASCIZ /STATUS VALID NOT RESET ON DRIVE RELEASE/
12941	055572	020123	040526	044514	
12942	055600	020104	047516	020124	
12943	055606	042522	042523	020124	
12944	055614	047117	042040	044522	
12945	055622	042526	051040	046105	
12946	055630	040505	042523	000	
12947	055635	105	050130	041505	EM50: .ASCIZ /EXPECTED 2ND INTERRUPT DID NOT OCCUR OR WAS LATE. COMMAND WAS: /
12948	055642	042524	020104	047062	
12949	055650	020104	047111	042524	
12950	055656	051122	050125	020124	
12951	055664	044504	020104	047516	

H03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 239
ERROR MESSAGES

SEQ 0239

12952	055672	020124	041517	052503	
12953	055700	020122	051117	053440	
12954	055706	051501	046040	052101	
12955	055714	027105	041440	046517	
12956	055722	040515	042116	053440	
12957	055730	051501	000072		
12958	055734	040503	047116	052117	EMS1: .ASCII /CANNOT READ BAD SECTOR FILE/<15><12>
12959	055742	051040	040505	020104	
12960	055750	040502	020104	042523	
12961	055756	052103	051117	043040	
12962	055764	046111	006505	012	
12963	055771	124	051505	044524	.ASCIZ /TESTING ABORTED ON THIS DRIVE/
12964	055776	043516	040440	047502	
12965	056004	052122	042105	047440	
12966	056012	020116	044124	051511	
12967	056020	042040	044522	042526	
12968	056026	000			
12969	056027	101	044514	047107	EMS2: .ASCII /ALIGNMENT PACK ON THIS DRIVE./<15><12>
12970	056034	042515	052116	050040	
12971	056042	041501	020113	047117	
12972	056050	052040	044510	020123	
12973	056056	051104	053111	027105	
12974	056064	005015			
12975	056066	042524	052123	047111	.ASCIZ /TESTING ABORTED ON THIS DRIVE./
12976	056074	020107	041101	051117	
12977	056102	042524	020104	047117	
12978	056110	052040	044510	020123	
12979	056116	051104	053111	027105	
12980	056124	000			
12981	056125	124	020117	040515	EMS3: .ASCIZ /TO MANY BAD SECTORS. TESTING ABORTED ON THIS DRIVE/
12982	056132	054516	041040	042101	
12983	056140	051440	041505	047524	
12984	056146	051522	020056	042524	
12985	056154	052123	047111	020107	
12986	056162	041101	051117	042524	
12987	056170	020104	047117	052040	
12988	056176	044510	020123	051104	
12989	056204	053111	000105		
12990	056210	042510	042101	051105	EMS4: .ASCIZ /HEADER WORD MISCOMPARE/
12991	056216	053440	051117	020104	
12992	056224	044515	041523	046517	
12993	056232	040520	042522	000	
12994	056237	104	052101	020101	EMS5: .ASCIZ /DATA WORD MISCOMPARE/
12995	056244	047527	042122	046440	
12996	056252	051511	047503	050115	
12997	056260	051101	000105		
12998					
12999	056264	040503	047116	052117	EMS6: .ASCIZ /CANNOT FIND HEADER 0 IN READ AND SORT HEADERS/
13000	056272	043040	047111	020104	
13001	056300	042510	042101	051105	
13002	056306	030040	044440	020116	
13003	056314	042522	042101	040440	
13004	056322	042116	051440	051117	
13005	056330	020124	042510	042101	
13006	056336	051105	000123		
13007	056342	052123	052101	051525	EMSVAL: .ASCIZ /STATUS VALID/

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 240
ERROR MESSAGES

SEQ 0240

13008	056350	053040	046101	042111	
13009	056356	000			
13010	056357	123	046105	041505	EMNZPR: .ASCIZ /SELECT WITH NON-ZERO STATUS PAIR/
13011	056364	020124	044527	044124	
13012	056372	047040	047117	055055	
13013	056400	051105	020117	052123	
13014	056406	052101	051525	050040	
13015	056414	044501	000122		
13016	056420	051127	052111	047111	EMWCS2: .ASCIZ /WRITING COMMAND STATUS REGISTER 2/
13017	056426	020107	047503	046515	
13018	056434	047101	020104	052123	
13019	056442	052101	051525	051040	
13020	056450	043505	051511	042524	
13021	056456	020122	000062		
13022		055036			EMDTPE= EM35 ;DRIVE TYPE ERROR
13023	056462	051104	053111	020105	EMDI: .ASCIZ /DRIVE INTERRUPT/
13024	056470	047111	042524	051122	
13025	056476	050125	000124		
13026		054753			EMOPAR= EM33 ;DRIVE DETECTED DRIVE BUS PARITY ERROR
13027	056502	051104	053111	020105	EMDSC: .ASCIZ /DRIVE STATUS CHANGE/
13028	056510	052123	052101	051525	
13029	056516	041440	040510	043516	
13030	056524	000105			
13031	056526	051104	053111	020105	EMDA: .ASCIZ /DRIVE ATTENTION/
13032	056534	052101	042524	052116	
13033	056542	047511	000116		
13034	056546	047503	052116	047522	EMCCLR: .ASCIZ /CONTROLLER CLEAR/
13035	056554	046114	051105	041440	
13036	056562	042514	051101	000	
13037		047640			EMSELD= OPER00 ;DRIVE SELECT
13038		047666			EMDCLR= OPER04 ;DRIVE CLEAR
13039		047727			EMRCAL= OPER12 ;RECALIBRATE
13040	056567	123	041505	047117	EM2INT: .ASCIZ /SECOND INTERRUPT/
13041	056574	020104	047111	042524	
13042	056602	051122	050125	000124	
13043	056610	042523	045505	052040	EMSKSF: .ASCIZ /SEEK TO SELF/
13044	056616	020117	042523	043114	
13045	056624	000			
13046		047752			EMSK= OPER16 ;SEEK
13047	056625	125	042516	050130	EMUXIT: .ASCIZ /UNEXPECTED INTERRUPT/
13048	056632	041505	042524	020104	
13049	056640	047111	042524	051122	
13050	056646	050123	000124		
13051	056652	047125	041111	051525	EMUR: .ASCIZ /UNIBUS RESET/
13052	056660	051040	051505	052105	
13053	056666	000			
13054	056667	122	051505	052105	EMRSET: .ASCIZ /RESET/
13055	056674	000			
13056		054527			EMOLT= EM26 ;DATA LATE
13057	056675	122	040505	044504	EMRDB: .ASCIZ /READING DATA BUFFER/
13058	056702	043516	042040	052101	
13059	056710	020101	052502	043106	
13060	056716	051105	000		
13061	056721	103	047117	051124	EMCERR: .ASCIZ /CONTROLLER ERROR/
13062	056726	046117	042514	020122	
13063	056734	051105	047522	000122	

J03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 241
ERROR MESSAGES

SEQ 0241

13064 056742 052523 051502 051531
13065 056750 042524 020115 046103
13066 056756 040505 000122
13067 056762 052515 052114 050111
13068 056770 042514 044440 052116
13069 056776 051105 052522 052120
13070 057004 000123
13071 057006 040502 020104 042523
13072 057014 052103 051117 020123
13073 057022 047117 050040 041501
13074 057030 020113 047111 040440
13075 057036 042522 051501 052440
13076 057044 042523 020104 054502
13077 057052 052040 051505 020124
13078 057060 041450 046131 031440
13079 057066 031061 034050 024451
13080 057074 005015
13081 057076 042524 052123 047111
13082 057104 020107 041101 051117
13083 057112 042524 020104 047117
13084 057120 052040 044510 020123
13085 057126 051104 053111 000105
13086
13087
13088 057134 051524 020124 052516
13089 057142 020115 051105 020122
13090 057150 041520 020040 051104
13091 057156 053111 000105
13092 057162 051524 020124 052516
13093 057170 020115 051105 020122
13094 057176 041520 020040 051104
13095 057204 053111 000105
13096 057210 045522 051503 020061
13097 057216 020040 045522 051503
13098 057224 020062 020040 045522
13099 057232 051504 020040 020040
13100 057240 045522 051105 020040
13101 057246 020040 045522 051501
13102 057254 043117 020040 045522
13103 057262 041504 046131 020040
13104 057270 045522 040504 000
13105 057275 122 041113 020101
13106 057302 020040 051040 053513
13107 057310 000103
13108 057312 030101 020060 020040
13109 057320 020040 030102 020060
13110 057326 020040 020040 030101
13111 057334 020061 020040 020040
13112 057342 030102 020061 020040
13113 057350 020040 030101 020062
13114 057356 020040 020040 030102
13115 057364 020062 020040 020040
13116 057372 030101 020063 020040
13117 057400 020040 030102 000063
13118 057406 045522 051503 020061
13119 057414 020040 045522 051503

EMSCLR: .ASCIZ /SUBSYSTEM CLEAR/

EMMI: .ASCIZ /MULTIPLE INTERRUPTS/

DRVABT: .ASCII /BAD SECTORS ON PACK IN AREAS USED BY TEST (CYL 312(8))<15><12>

.ASCIZ /TESTING ABORTED ON THIS DRIVE/

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

DH002A: .ASCIZ /TST NUM ERR PC DRIVE/

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

DH002C: .ASCIZ /RKBA RKWC/

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

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

K03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 242
DATA HEADERS FOR ERROR REPORTS

SEQ 0242

13120	057422	020062	020040	045522
13121	057430	051504	020040	020040
13122	057436	045522	051105	020040
13123	057444	020040	045522	051501
13124	057452	043117	020040	045522
13125	057460	051115	000061	
13126	057464	044124	020105	041101
13127	057472	053117	020105	051101
13128	057500	020105	054105	042520
13129	057506	052103	042105	042440
13130	057514	051122	051117	020123
13131	057522	044124	052101	042040
13132	057530	042111	047040	052117
13133	057536	051440	052105	044440
13134	057544	020116	050117	051105
13135	057552	052101	047511	035116
13136	057560	000		
13137	057561	124	042510	040440
13138	057566	047502	042526	040440
13139	057574	042522	052440	042516
13140	057602	050130	041505	042524
13141	057610	020104	051105	047522
13142	057616	051522	051440	052105
13143	057624	044440	020116	050117
13144	057632	051105	052101	047511
13145	057640	035116	000	
13146	057643	124	042510	040440
13147	057650	047502	042526	040440
13148	057656	042522	042440	051122
13149	057664	051117	020123	042523
13150	057672	020124	047111	047440
13151	057700	042520	040522	044524
13152	057706	047117	000072	
13153	057712	047101	020131	044506
13154	057720	046105	020104	044527
13155	057726	044124	040440	046114
13156	057734	047440	042516	020123
13157	057742	052515	052123	041040
13158	057750	020105	047503	051516
13159	057756	042111	051105	042105
13160	057764	044440	053116	046101
13161	057772	042111	000	
13162	057775	105	051122	051117
13163	060002	040440	020124	047503
13164	060010	050115	042514	044524
13165	060016	047117	047440	020106
13166	060024	050117	051105	052101
13167	060032	047511	000116	
13168	060036	054105	052120	020104
13169	060044	020040	051511	000
13170	060051	122	040505	020104
13171	060056	040504	040524	053440
13172	060064	052111	020110	047506
13173	060072	041522	042105	041440
13174	060100	046131	047111	042504
13175	060106	020122	053117	051105

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

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

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

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

DH010: .ASCIZ /ERROR AT COMPLETION OF OPERATION/

DH010A: .ASCIZ /EXPTD IS/

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

L03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 243
DATA HEADERS FOR ERROR REPORTS

SEQ 0243

13176	060114	046106	053517	000
13177	060121	116	052117	051440
13178	060126	052105	040440	020123
13179	060134	020101	042522	052523
13180	060142	052114	047440	000106
13181	060150	047516	020124	042522
13182	060156	042523	020124	051501
13183	060164	040440	051040	051505
13184	060172	046125	020124	043117
13185	060200	000		
13186	060201	123	052105	040440
13187	060206	020123	020101	042522
13188	060214	052523	052114	047440
13189	060222	000106		
13190	060224	042522	042523	020124
13191	060232	051501	040440	051040
13192	060240	051505	046125	020124
13193	060246	043117	000	
13194	060251	107	047517	020104
13195	060256	020040	041040	042101
13196	060264	020040	020040	053440
13197	060272	051117	020104	052516
13198	060300	000115		
13199	060302	052523	051502	051531
13200	060310	042524	020115	046103
13201	060316	040505	020122	047524
13202	060324	051040	051505	052105
13203	060332	051440	042505	020113
13204	060340	044514	044515	020124
13205	060346	046101	047514	044527
13206	060354	043516	044040	040505
13207	060362	051504	052040	020117
13208	060370	042522	047514	042101
13209	060376	000		
13210	060377	123	046105	041505
13211	060404	020124	043101	042524
13212	060412	020122	042522	040503
13213	060420	000114		
13214	060422	042522	040503	020114
13215	060430	047503	050115	042514
13216	060436	042524	040440	052124
13217	060444	020116	043101	042524
13218	060452	020122	042523	042514
13219	060460	052103	000	
13220				
13221				
13222		060464		
13223	060464	001302	001116	001626
13224	060472	001540	001550	001552
13225	060500	001554	001556	001566
13226	060506	001302	001116	001626
13227	060514			
13228	060514			
13229	060514			
13230	060514			
13231	060514			

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

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

DH013: .ASCIZ /SET AS A RESULT OF/

DH014: .ASCIZ /RESET AS A RESULT OF/

DH015: .ASCIZ /GOOD BAD WORD NUM/

DH016: .ASCIZ /SUBSYSTEM CLEAR TO RESET SEEK LIMIT ALLOWING HEADS TO RELOAD/

DH017: .ASCIZ /SELECT AFTER RECAL/

DH018: .ASCIZ /RECAL COMPLETE ATTN AFTER SELECT/

.SBTTL DATA TABLES FOR ERROR REPORTS

DT003: .EVEN
.WORD \$TESTN,\$ERRPC,\$RVNUM,\$T.CS1,\$T.CS2,\$T.DS,\$T.ER,\$T.ASOF,\$T.MR1

DT001: .WORD \$TESTN,\$ERRPC,\$RVNUM

DT002:
DT004:
DT005:
DT006:
DT007:

M03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 244
DATA TABLES FOR ERROR REPORTS

SEQ 0244

13232	060514	001302	001116	001626	DT010:	.WORD	\$TESTN,\$ERRPC,DRVNUM
13233	060522	001540	001550	001552		.WORD	T.CS1,T.CS2,T.DS,T.ER,T.ASOF
13234	060530	001554	001556				
13235	060534	001560	001546			.WORD	T.DCYL,T.DA
13236	060540	001544	001542			.WORD	T.BA,T.WC
13237	060544	001202	001204	001206		.WORD	\$REG10,\$REG11,\$REG12,\$REG13,\$REG14,\$REG15,\$REG16,\$REG17
13238	060552	001210	001212	001214			
13239	060560	001216	001220				
13240	060564	001302	001116	001626	DT015:	.WORD	\$TESTN,\$ERRPC,DRVNUM
13241	060572	001202	001204	001206	DT015A:	.WORD	\$REG10,\$REG11,\$REG12
13242					.SBTTL	DATA FORMATS FOR ERROR REPORTS	
13243	060600	000001			DF001:	.WORD	1
13244	060602	003	000			.BYTE	3,0
13245							
13246	060604	000004			DF002:	.WORD	4
13247	060606	000	000			.BYTE	0,0
13248	060610	057162				.WORD	DH002A
13249	060612	003	000			.BYTE	3,0
13250	060614	057210				.WORD	DH002B
13251	060616	007	000			.BYTE	7,0
13252	060620	057275				.WORD	DH002C
13253	060622	002	000			.BYTE	2,0
13254							
13255							
13256	060624	000002			DF003:	.WORD	2
13257	060626	003	000			.BYTE	3,0
13258	060630	057406				.WORD	DH003B
13259	060632	006	000			.BYTE	6,0
13260							
13261	060634	000006			DF004:	.WORD	6
13262	060636	000	000			.BYTE	0,0
13263	060640	000000			DF004A:	.WORD	0
13264	060642	000	000			.BYTE	0,0
13265	060644	057162				.WORD	DH002A
13266	060646	003	000			.BYTE	3,0
13267	060650	057210				.WORD	DH002B
13268	060652	007	000			.BYTE	7,0
13269	060654	057275				.WORD	DH002C
13270	060656	002	000			.BYTE	2,0
13271	060660	057312				.WORD	DH002D
13272	060662	010	000			.BYTE	10,0
13273							
13274	060664	000007			DF005:	.WORD	7
13275	060666	000	000			.BYTE	0,0
13276	060670	000000			DF005A:	.WORD	0
13277	060672	000	000			.BYTE	0,0
13278	060674	057162				.WORD	DH002A
13279	060676	003	000			.BYTE	3,0
13280	060700	057210				.WORD	DH002B
13281	060702	007	000			.BYTE	7,0
13282	060704	057275				.WORD	DH002C
13283	060706	002	000			.BYTE	2,0
13284	060710	057312				.WORD	DH002D
13285	060712	010	000			.BYTE	10,0
13286	060714	057712				.WORD	DH005A
13287	060716	000	000			.BYTE	0,0

N03

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(100%) 31-JAN-77 16:34 PAGE 245
DATA FORMATS FOR ERROR REPORTS

SEQ 0245

13288
13289 060720 000005
13290 060722 000 000
13291 060724 000000
13292 060726 000 000
13293 060730 057162
13294 060732 003 000
13295 060734 057210
13296 060736 007 000
13297 060740 057275
13298 060742 002 000
13299
13300 060744 000004
13301 060746 000 000
13302 060750 000000
13303 060752 000 000
13304 060754 057162
13305 060756 003 000
13306 060760 057406
13307 060762 006 000
13308
13309 060764 000004
13310 060766 000 000
13311 060770 000000
13312 060772 000 000
13313 060774 057162
13314 060776 003 000
13315 061000 060036
13316 061002 002 000
13317
13318 061004 000004
13319 061006 000 000
13320 061010 000000
13321 061012 000 000
13322 061014 057162
13323 061016 003 000
13324 061020 057406
13325 061022 000006 000000
13326
13327 061026 000002
13328 061030 003 000
13329 061032 060251
13330 061034 003 000
13331
13332 061036 000001
13333 061040 003 000
13334
13335 061042 000052
13336 061166 000052
13337 061376 000240
13338 061376 000240
13339 061400 005014
13340 061402 005237 001662
13341 061406 000240
13342 061410 000002
13343 061412 000240

DF006: .WORD 5
.BYTE 0,0
DF006A: .WORD 0
.BYTE 1,0
.WORD D4002A
.BYTE 3,0
.WORD D4002B
.BYTE 7,0
.WORD D4002C
.BYTE 2,0

DF007: .WORD 4
.BYTE 0,0
DF007A: .WORD 0
.BYTE 0,0
.WORD D4002A
.BYTE 3,0
.WORD D4003B
.BYTE 6,0

DF010: .WORD 4
.BYTE 0,0
DF010A: .WORD 0
.BYTE 0,0
.WORD D4002A
.BYTE 3,0
.WORD D4010A
.BYTE 2,0

DF011: .WORD 4
.BYTE 0,0
DF011A: .WORD 0
.BYTE 0,0
.WORD D4002A
.BYTE 3,0
.WORD D4003B
.WORD 6,0

DF015: .WORD 2
.BYTE 3,0
.WORD D4015
.BYTE 3,0

DF016: .WORD 1
.BYTE 3,0

BS26: .BLKW 52
BS24: .BLKW 52
. =61376
NOP
SPCHLR: CLR (R4)
INC INTSET
NOP
SPCPAR: RTI
NOP

;CLEAR MEM MAINT REG
;COUNT THE INTERRUPT
;RETURN

B04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 246
DZR6KD.P11 31-JAN-77 16:26 DATA FORMATS FOR ERROR REPORTS

SEQ 0246

13344 061414 001000
13345 063414 001000
13346 000001

IBUFF: .BLKW 1000
OBUFF: .BLKW 1000
.END

C04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 248
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0247

ABASE = 177440	1955#	2277	2318
ABORT 053215	9529	12702#	
ABTFAI 043650	9324	11293#	
ACDW1 = 000000	2277	2320	
ACDW2 = 000000	2277		
ACLO = 000010	2050#	10387	
ACLOER= 000100	10389	12258#	
ACPUOP= 000000	2277	2292	
ADWD0 = 000000	2277		
ADWD1 = 000000	2277		
ADWD10= 000000	2277		
ADWD11= 000000	2277		
ADWD12= 000000	2277		
ADWD13= 000000	2277		
ADWD14= 000000	2277		
ADWD15= 000000	2277		
ADWD2 = 000000	2277		
ADWD3 = 000000	2277		
ADWD4 = 000000	2277		
ADWD5 = 000000	2277		
ADWD6 = 000000	2277		
ADWD7 = 000000	2277		
ADWD8 = 000000	2277		
ADWD9 = 000000	2277		
ADEVCT= 000000	2277	2283	
ADEVN = 000000	2277	2319	
ADJCLK 043274	3207	11186#	
RENV = 000000	2277	2288	
RENVN = 000000	2277	2289	
AFATAL= 000000	2277	2280	
AMADR1= 000000	2277	2305	
AMADR2= 000000	2277	2309	
AMADR3= 000000	2277	2312	
AMADR4= 000000	2277	2315	
AMAMS1= 000000	2277	2299	
AMAMS2= 000000	2277	2307	
AMAMS3= 000000	2277	2310	
AMAMS4= 000000	2277	2313	
AMSGAD= 000000	2277	2285	
AMSGLG= 000000	2277	2286	
AMSGTY= 000000	2277	2279	
AMTYP1= 000000	2277	2300	
AMTYP2= 000000	2277	2308	
AMTYP3= 000000	2277	2311	
AMTYP4= 000000	2277	2314	
APASS = 000000	2277	2282	
APRIOR= 000240	1954#	2277	
APTCSU= 000040	9382#	11326	
APTEMV= 000001	9338	9380#	9437 11319
APTSIZ= 000200	2780	9379#	
APTSP0= 000100	9340	9381#	11321
ASWREG= 000000	2277	2290	
ATESTN= 000000	2277	2281	
AUNIT = 000000	2277	2284	
AUSWR = 000000	2277	2291	
AVECT1= 000210	1953#	2277	2316

SEQ 0248

[illegible]

SEQ 0249

BS26CT	001646	2679#	4063*
CCLR =	100000	2007#	3448
COT =	002000	2000#	3358
CERNER=	100000	10326	12267#
CERR =	100000	2006#	3037
CFMT =	010000	2003#	6409
		10950	10957
CHKBA	037110	10155#	10349
CHKCTS	037322	6788	10217#
CHKOP	037642	10307#	11965
CHKST	037650	10305	10310#
CHKWC	037056	10136#	10347
CHKWE	037610	10298#	11966
CKSWR =	104407	9185	9398
CLEAR =	000105	1980#	3536
CLKCAL	035102	9684#	9779
CMNDLB	047600	9790	10353
COE =	001000	2037#	
COERR =	001000	6384	12261#
CP1170=	002000	2696#	5512
CR =	000015	1818#	11365
CRLF =	000200	1819#	2804
CSRPTR	001672	2702#	5514*
		6623#	6628*
		10316	12230#
CS1ERB=	024000	10318	12231#
CS2ERB=	177400	2001#	10048
CTO =	004000	7785	10440
CTOERR=	000002	2423	2952
CTRHLT	043404	10267	12292#
CYLERR=	000010	2043#	10381
DCK =	100000	8011	10411
DCKERR=	000002	1825#	2224
DOISP =	177570	2055#	
DOT =	000400	2709#	10966
DESHLD	001700	2371	13243#
DF001	060600	2386	13246#
DF002	060604	2397	13256#
DF003	060624	2478	13261#
DF004	060634	10577	10588
DF004A	060640	2492	13274#
DF005	060664	10597	13276#
DF005A	060670	2517	13289#
DF006	060720	10555	10567
DF006A	060724	2543	13300#
DF007	060744	10529	13302#
DF007A	060750	2566	13309#
DF010	060764	5398#	5451*
DF010A	060770	10353#	13311#
		2580	2589
DF011	061004	3193#	3276*
DF011A	061010	3487#	3493*
		3796#	3816*
		2618	13327#
DF015	061026	2626	13332#
DF016	061036	2369	13088#
DM001	057134		

SEQ 0250

DH002A	057162	2395	2616	13092#	13248	13265	13278	13293	13304	13313	13322						
DH002B	057210	13096#	13250	13267	13280	13295											
DH002C	057275	13105#	13252	13269	13282	13297											
DH002D	057312	13108#	13271	13284													
DH003B	057406	13118#	13258	13306	13324												
DH005	057464	10638	13126#														
DH005A	057712	13153#	13286														
DH006	057561	10641	10698	13137#													
DH007	057643	10636	13146#														
DH010	057775	2564	13162#														
DH010A	060036	13168#	13315														
DH010B	060051	13170#															
DH011	060121	2578	13177#														
DH012	060150	2587	13181#														
DH013	060201	2596	13186#														
DH014	060224	2604	13190#														
DH015	060251	13194#	13329														
DH016	060302	7900	13199#														
DH017	060377	8775	13210#														
DH018	060422	8783	13214#														
DH2N	001372	2384#	7900#	8775*	9790*	9957*	9995*										
DH4N	001412	2476#	10578	10589													
DH5N	001422	2490#	10598														
DH6N	001432	2515#	10556	10568													
DH7N	001442	2541#	10530														
DI =	040000	2005#	3427	3461	3484	3524	3556	3752	3793	3820	8913	8932	10102				
DISPLA	001142	2224#	2768#	2776*	9245*	9401*											
DISPRE	000174	2165#	2776														
DLT =	100000	2024#	4161	10416													
DLTERR=	000010	10418	12273#														
DMD =	000040	2066#	3589	3597	3636	3645	8072	8154	8162	8262	8385	11197					
DRA =	000001	2047#															
DRDY =	000200	2054#	3051	3056	3080	8523											
DRDT =	000040	2052#	10395														
DRDTER=	000400	8290	8424	10397	12260#												
DRPAR =	000010	2031#															
DRPARE=	000010	3409	12255#														
DRSTER=	040000	10593	10599	10733	12344#												
DRVABT	057006	4089	13071#														
DRVBIT	001630	2667#	3132*	3197	3199*	3998	3999	4048	4049	8550	8554*	8555	8557	8563*			
DRVDRP	001666	2700#	2881*	3023*	3047*	3084*	3134	3164	3999*	4049*							
DRVMSK=	000007	2011#															
DRVNUM	001626	2666#	3141*	3203*	3223	3263	3313	3330	3356	3400	3416	3578	3626	3676			
		3742	3764	3848	3865	3909	3926	6410	6438	6898	7410	8521	8553*	8564*			
		8741*	8753*	8774*	8894*	8915*	8935*	8944*	8959*	9823	9949	11189	13223	13226			
		13232	13240														
DRVAPT=	004000	2697#	2882	3014	3101	3119											
DSC =	040000	2058#	3190	3433	3490	3706											
DSERII=	000070	10320	12234#														
DSWR =	177570	1824#	2223	2767													
DTE =	010000	2040#															
DTERR =	010000	8083	12264#														
DTYE =	000040	2033#															
DTYERP=	000040	12257#															
DTCC1	060506	2370	13226#														
DT002	060514	2385	13227#														

G04

AK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 252
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0251

DT003	060464	2396	13223#																
DT004	060514	2477	13228#																
DT005	060514	2491	13229#																
DT006	060514	2516	13230#																
DT007	060514	2542	13231#																
DT010	060514	2579	2588	2597	2605	13232#													
DT015	060564	2565	2617	13240#															
DT015A	060572	2625	13241#																
DOTST =	000001	2689#	3024	3046	3058	3064													
ECCW =	020000	2074#																	
ECH =	000100	2034#	10381	10405															
ECERR=	000001	8011	10407	12270#															
EMCCLR	056546	3458	3464	13034#															
EMCERR	056721	4170	13061#																
EMDA	056526	3424	3457	3480	3520	3552	8629	8695	8962	13031#									
ENDCLR=	047666	3553	3559	13038#															
ENDI	056462	3429	3463	3486	3526	3558	3754	3795	3822	13023#									
ENDLT =	054527	4163	10008	13056#															
ENDPAR=	054753	3425	3430	3436	3442	3481	3487	3493	3499	13026#									
ENDSC	056502	3192	3435	3492	3708	13027#													
ENDTPE=	055036	3374	13022#																
EHMI	056762	3766	13067#																
EHMZPR	056357	3292	3521	3527	13010#														
EHRCAL=	047727	3687	13039#																
EMRDB	056675	4164	4171	10009	13057#														
EMRSET	056667	3816	3823	13054#															
EMSCLR	056742	3193	8696	13064#															
EMSELD=	047640	3276	3327	3726	13037#														
EMSK =	047752	8963	13046#																
EMSKSF	056610	3755	3767	3796	13043#														
EMSVL	056342	3275	3291	3326	3337	3373	3441	3498	3686	3701	3725	8765	8782	13007#					
EMTVEC=	000030	1913#	2751*	2752*															
EMUR	056652	8630	13051#																
EMUXIT	056625	3815	13047#																
EMWCS2	056420	3338	13016#																
EM1	053303	2921	2950	12714#															

H04

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 253
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0252

EM2	053363	12723#																
EM2INT	056567	3702	3709	13040#														
EM2N	001370	2383#	3695*	9791*	9966*	9994*												
EM20	054376	12308	12818#															
EM21	054422	12309	12822#															
EM22	054446	12310	12826#															
EM23	054467	12312	12829#															
EM24	054500	12313	12831#															
EM25	054513	12314	12833#															
EM26	054527	12315	12836#	13056														
EM27	054541	12316	12838#															
EM28	054566	12317	12842#															
EM29	054601	12318	12844#															
EM3	053444	9562	12732#															
EM3N	001400	2394#	3158*	3172*	3186*	3243*	10108*	10110*	10121*									
EM30	054622	12320	12847#															
EM31	054675	12321	12855#															
EM32	054715	12322	12858#															
EM33	054753	12323	12864#	13026														
EM34	055021	12324	12871#															
EM35	055036	12325	12874#	13022														
EM36	055057	12326	12877#															
EM37	055066	12327	12879#															
EM38	055111	12328	12883#															
EM39	055131	12329	12886#															
EM4	053514	9791	9966	12739#														
EM4N	001410	2475#	3520*															
EM40	055153	12330	12890#															
EM41	055205	12331	12895#															
EM42	055226	12332	12898#															
EM43	055251	12333	12902#															
EM44	055313	12334	12908#															
EM45	055330	12335	12911#															
EM46	055370	3186	12918#															
EM47	055425	3158	12923#															
EM48	055506	3172	12932#															
EM49	055565	3243	12940#															
EM5	053607	10108	12749#															
EM5N	001420	2489#																
EM50	055635	3695	12947#															
EM51	055734	3996	4007	4017	4024	12958#												
EM52	056027	4033	12969#															
EM53	056125	4046	12981#															
EM54	056210	10879	12990#															
EM55	056237	10882	12994#															
EM56	056264	9994	12999#															
EM6	053653	10110	12756#															
EM6N	001430	2514#																
EM7	053732	10121	12764#															
EM7N	001440	2540#																
ERRCNT	001632	2668#	9524*	9527														
ERRLMT	001634	2670#	4193	4689	4762	4843	4912	4993	5062	5142	5213	5287	5358	5694				
		5782	5865	5950	6035	6094	6151	6209	6265	6321	6455	6530	6707	6832				
		6870	7507															
ERRVEC=	000004	1906#	2765	2766*	2777*	9113	9114	9118*	9123*	9135*	9140*	9149*	9159*	9160*				
		9196	9197*	9199*	9202*	9582	9583	9584*	9585*	9619*	9642*	9657*	9658*					

[illegible]

AK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 255
CROSS REFERENCE TABLE -- USER SYMBOLS

		7444*	7599*	7659*	7737*	7789*	7826*	7895*	7990	8049*	8092*	8196*	8232*	8329*
		8456*	8595*	8663*	8750*	8769*	8901*	8922*		9553*	9643*	9651	9656*	9689
		9712	9780	9835*	9954*	9958	11238*	13340*						
		1911*	2749*	2750*										
IOTVEC=	000020													
IR	= 000100													
IWAT1M	035224	9736*	11980											
IWAT1S	035244	9742*	11977											
IWAT11	035304	9754*	11973											
IWAT12	035274	9751*	11974											
IWAT14	035264	9748*	11975											
IWAT15	035254	9745*	11976											
IWAT16	035364	9772*	11967											
IWAT2S	035234	9739*	11978											
IWAT32	035354	9769*	11968											
IWAT48	035344	9766*	11969											
IWAT64	035334	9763*	11970											
IWAT8S	035214	9733*	11979											
IWAT80	035324	9760*	11971											
IWAT96	035314	9757*	11972											
KIPAR0=	172340	1944*	9099	9127										
KIPAR1=	172342	1945*												
KIPAR2=	172344	1946*												
KIPAR3=	172346	1947*												
KIPAR4=	172350	1948*												
KIPAR5=	172352	1949*												
KIPAR6=	172354	1950*	11026*											
KIPAR7=	172356	1951*	9143											
KIPDR0=	172300	1933*												
KIPDR1=	172302	1934*												
KIPDR2=	172304	1935*												
KIPDR3=	172306	1936*												
KIPDR4=	172310	1937*												
KIPDR5=	172312	1938*												
KIPDR6=	172314	1939*												
KIPDR7=	172316	1940*												
KHLAD0	001710	2713*	5545*	5561*	5616*	6599*	6615*	6643*	9573*	9647*	9939*	9985*	11195*	
KHLVEC	001712	2714*	9644											
LCKHLR	034420	9572*	9645											
LCLKPR=	100000	2698*	9654	9685	9782	9937	9983	11193						
LCLKTK	001674	2704*	3804*	3806	9572*	9684*	9691	9693*	9718*	9785	9955*	9961		
LF	= 000012	1817*	11369	11375										
LOADRK	035570	9835*	11963											
LRLAD0	035530	3974	4112	4140	4237	4265	4313	4346	4392	4412	4444	4465	4498	4526
		4559	4585	4620	4649	4668	4714	4734	4801	4825	4876	4893	4955</	

SEQ 0255

		6409*	6437*	6899*	8589*	8658*	8742*	8754*	8896*	8906*	8917*	8925*	8938*	8949*
L.CS2	001610	9818	9850	9941	9956	10094	10164	10241	10898	10926	10950	10957		
		2654*	3027*	3070*	3140*	3223*	3233*	3263*	3313*	3356*	3400*	3578*	3588*	3626*
		3637*	3676*	3742*	3784*	3848*	3865*	3909*	3926*	4722*	6410*	6438*	6898*	7464*
		7479*	7564*	7635*	7691*	8155*	8588*	8657*	8740*	8752*	8895*	8916*	8936*	8945*
		10040	10158											
L.DA	001606	2651*	9951											
L.DCYL	001614	2656*	3928*	6414*	6440*	6903*	8937*	8946*	9950	10246	10767	10890	11262*	
L.DS	001606	2652*	6936*	10249										
L.DT	001607	2653*	3867*	4204	6412*	6439*	6900*	10247*	10770	10889	11273	11275*		
L.MR1	001616	2657*	3265*	3279*	3402*	3446*	3510*	3534*	3589*	3636*	7840*	7879*	8072*	8154*
		8262*	8283*	8385*	8417*	8440*	8468*							
L.WC	001602	2649*	6411*	6902*	8947*	9838	10162	10213						
MCCLK =	000400	2069*	9915	9916										
MCLOCK	036004	3593	3641	8075	8159	8266	8388	8392	8398	9906*				
MDS =	001000	2018*	10446											
MDSERR=	000010	10448	12282*											
MEMBAS	001714	2715*	9596											
MEMPAR	001670	2701*	9586*	9600*	9633									
MEMPYB=	000004	2691*	5497	5500										
MEMSZB=	000002	2690*	4788	4793	4936	4942	5087	5093						
MEMD =	001000	2070*												
MESMSK=	000017	2063*												
MEMD =	002000	2071*												
MILCNT	001652	2681*	9711											
MIND =	000200	2068*	8391	8396										
MIVC =	000250	1922*												
MIVCA	001716	2716*	5539*	5615*	6593*	6642*	9603	9632						
MSP =	000100	2067*												
MYTIME	035150	3805	9710*	9784	9960									
M.CAD=	017760	2138*												
M.COIF=	017760	2137*												
M.DRV =	000007	2134*												
M.HEAD=	007000	2141*												
M.ID =	000003	2136*												
M.PAR =	100000	2135*												
M.SET=	000760	2140*												
M.SER =	077770	2139*												
NCERME=	020000	10401	12265*											
NED =	010000	2021*	3039	3095	10434									
NEDEKR=	000001	10436	12279*											
NEM =	004000	2020*	10454											
NEMERR=	000040	5395	5448											

SEQ 0256

[illegible]

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRBKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 258
CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

SEQ 0258

[illegible]

AK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZRGKD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 260
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0259

STKLMT=	177774	1822#												
SVAL =	100000	2059#	3156	3170	3241	3272	3288	3323	3334	3370	3439	3496	3684	3699
		3723#	8763	8779										
SWR	001140	2223#	2745	2767*	2769	2775*	2782*	2796	9191	9205	9208	9223	9230	9402
		9409	9444	9448	9525	9560	9672	10628	11564	11603	11658*	11885	11898*	
SWREG	000176	2166#	2775	2796	11564	11603	11626							
SW0	= 000001	1875#												
SW00	= 000001	1865#	1875											
SW01	= 000002	1864#	1874											
SW02	= 000004	1963#	1873											
SW03	= 000010	1862#	1872											
SW04	= 000020	1861#	1871											
SW05	= 000040	1860#	1870											
SW06	= 000100	1859#	1869											
SW07	= 000200	1858#	1868											
SW08	= 000400	1857#	1867											
SW09	= 001000	1856#	1866											
SW1	= 000002	1874#												
SW10	= 002000	1855#												
SW11	= 004000	1854#												
SW12	= 010000	1853#	9525											
SW13	= 020000	1852#	10628											
SW14	= 040000	1851#												
SW15	= 100000	1850#												
SW2	= 000004	1873#												
SW3	= 000010	1872#												
SW4	= 000020	1871#												
SW5	= 000040	1870#												
SW6	= 000100	1869#												
SW7	= 000200	1868#												
SW8	= 000400	1867#												
SW9	= 001000	1866#	9672											
S.ACLO=	000100	2094#												
S.BRMH=	000100	2108#												
S.BRKE=	040000	2130#												
S.CART=	000400	2110#												
S.DIB =	002000	2126#												
S.DOOR=	000200	2109#												
S.DRA =	000040	2080#												
S.DROT=	020000	2101#												
S.DRY =	000200	2082#												
S.DSC =	040000	2089#												
S.FLT =	000200	2095#												
S.FORM=	001000	2084#												
S.FWD =	002000	2112#												
S.HDFL=	000200	2123#												
S.HDWM=	000040	2107#	7887											

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 261
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0260

S. PLO = 004000	2127#												
S. REV = 004000	2113#												
S. RTZ = 020000	2115#												
S. SECT = 000020	2120#												
S. SKI = 002000	2098#												
S. SPIN = 010000	2087#												
S. SPLS = 010000	2100#												
S. SPOK = 001000	2111#												
S. TYPE = 000400	2083#												
S. UNLD = 040000	2116#												
S. UNS = 040000	2102#												
S. VV = 000100	2081#												
S. WCLK = 000040	2121#												
S. WGAT = 000100	2122#												
S. WLE = 004000	2099#												
S. WRL = 004000	2086#												
S. XDOK = 000020	2106#												
S. XERR = 001000	2125#												
TBITVE = 000014	1908#												
TCHKOP = 104421	3042	3098	3167	3181	3230	3239	3269	3285	3320	3452	3515	3542	3585
	3633	3719	3770	3985	4131	4156	4182	4260	4293	4339	4373	4409	4462
	4523	4582	4596	4635	4665	4680	4731	4754	4817	4835	4891	4904	4967
	4984	5039	5053	5117	5132	5190	5204	5253	5273	5327	5344	5651	5670
	5685	5735	5758	5773	5821	5845	5906	5926	5941	5992	6011	6026	6085
	6142	6200	6256	6312	6356	6423	6443	6506	6521	6634	6683	6698	6759
	6823	6912	6942	6985	7000	7042	7057	7178	7193	7307	7324	7371	7402
	7426	7445	7471	7496	7585	7601	7607	7622	7647	7661	7666	7681	7704
	7734	7740	7755	7830	7884	7973	8017	8108	8141	8167	8213	8229	8237
	8252	8326	8333	8348	8363	8406	8447	8482	8500	8544	8612	8680	8747
	8760	8911	8930	8956	11235	11242	11270	11965#					
TCHKWE = 104422	3150	3364	3408	3473	3602	3650	3878	3886	3934	4280	4361	4424	4477
	4538	5392	5445	5579	6383	6782	6858	7077	7215	7782	7860	7934	8009
	8082	8289	8423	11966#									
TGETRK = 104420	3036	3077	3332	3549	3683	3698	3750	3791	3819	4160	8624	8690	8778
	9787	9965	10011	10111	10310	11964#							
TIMCNT 001654	2682#	9687#	9716#	9719#	9778#	9953#							
TKVEC = 000060	1915#	11541#	11542#										
TLOADR = 104417	3031	3072	3144	3176	3226	3235	3266	3281	3316	3360	3404	3449	3469
	3504	3511	3538	3581	3591	3629	3639	3679	3715	3745	3787	3850	3869
	3911	3930	3982	4127	4152	4256	4273	4336	4354	4405	4420	4458	4473
	4519	4534	4578	4592	4631	4661	4676	4727	4750	4813	4832	4887	4901
	4963	4980	5035	5049	5113	5128	5186	5200	5249	5269	5323	5341	5389
	5442	5541	5647	5666	5681	5731	5754	5769	5817	5841	5855	5902	5922
	5937	5987	6007	6022	6081	6138	6196	6252	6308	6352	6379	6419	6502
	6517	6595	6679	6694	6755	6778	6819	6854	6908	6938	6981	6996	7038
	7053	7073	7174	7189	7211	7303	7320	7367	7398	7441	7466	7492	7565
	7596	7618	7637	7656	7677	7694	7730	7751	7766	7778	7822	7841	7856
	7880	7927	7969	7988	8005	8046	8061	8073	8105	8137	8157	8193	8209
	8225	8248	8264	8285	8322	8344	8359	8374	8387	8419	8443	8478	8496
	8540	8591	8659	8744	8756	8898	8907	8919	8926	8940	8951	11231	11266
	11963#												
TPVEC = 000064	1916#												
TRAPVE = 000034	1914#	2753#	2754#										
TRKERR = 000020	6789	10270	12293#										
TRTVEC = 000014	1909#												
TSSINI = 104416	2999	3067	3137	3220	3260	3310	3353	3397	3575	3608	3623	3673	3739

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 262
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0261

[illegible]

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 263
CROSS REFERENCE TABLE -- USER SYMBOLS

TST51	017132	5970#		9290															
TST52	017336	6057#		9291															
TST53	017462	6116#		9292															
TST54	017606	6173#		9293															
TST55	017732	6230#		9294															
TST56	020056	6286#		9295															
TST57	020202	6339#		9296															
TST6	004746	3308#		9255															
TST60	020246	6366#		9297															
TST61	020320	6401#		9298															
TST62	020576	6480#		9299															
TST63	020760	6554#		9300															
TST64	021372	6656#		9301															
TST65	021560	6735#		9302															
TST66	022264	6889#		9303															
TST67	022474	6964#		9304															
TST7	005074	3351#		9256															
TST70	022600	7020#		9305															
TST71	023222	7156#		9306															
TST72	023644	7286#		9307															
TST73	023754	7348#		9308															
TST74	024606	7550#		9309															
TST75	025264	7717#		9310															
TST76	025500	7808#		9311															
TST77	025756	7914#		9312															
TWAT1M=	104440	11980#																	
TWAT1S=	104435	11977#																	
TWAT1I=	104431	3983	4128	4257	4337	4406	4459	4520	4579	4632	4662	6756	6909	7368					
		11973#																	
TWAT12=	104432	11974#																	
TWAT14=	104433	11975#																	
TWAT15=	104434	6353	6380	6420	7399	7566	7638	7695	8457	11267	11976#								
TWAT16=	104423	3033	3074	3145	3178	3227	3236	3267	3282	3317	3361	3405	3470	3512					
		3539	3582	3599	3630	3647	3680	3716	3746	3788	3812	3852	3870	3913					
		3931	4153	7442	7597	7619	7657	7678	7731	7752	7767	7823	7842	7857					
		7881	7931	8047	8062	8106	8138	8194	8210	8226	8249	8272	8286	8323					
		8345																	

SEQ 0263

		11660	11679	11727	11733	11738	11742	11747	11748	11750	11753	11757	11823	11825
TYPERR	034050	11907	11947#											
TYPOC =	104402	9434	9470#											
TYPON =	104404	2836	2847	2864	9418	9423	9503	9563	11627	11948#				
TYPOS =	104403	11950#												
T.ASOF	001556	11949#												
		2637#	3422	3455	3478	3518	3550	8598*	8599	8626	8666*	8667	8692	8960
T.BA	001544	13223	13233											
		2632#	5406	5410	5460	5464	5582	5591	5595	5598	7095	7100	7114	7118
T.CS1	001540	7233	7238	7252	7256	10188	10191	10276	13236					
		2630#	3037	3078	3427	3461	3484	3524	3556	3752	3793	3820	4168	5399
		5452	8913	8932	9788	9863	10101*	10102	10106	10119	10192	10316	10324	10341
T.CS2	001550	10374	10642	13223	13233									
T.DA	001546	2634#	3039	3095	4161	10318	10375	13223	13233					
T.DB	001564	2633#	10268	10271	10278	10280	13235							
T.DCYL	001560	2640#												
T.DS	001552	2638#	10265	10277	13235									
		2635#	3051	3056	3080	3082	3156	3170	3184	3190	3241	3272	3288	3323
		3334	3370	3433	3439	3490	3496	3684	3699	3706	3723	8763	8779	10320
		10376	13223	13233										
T.ECPS	001570	2642#												
T.ECPT	001572	2643#												
T.ER	001554	2636#	10322	10377	13223	13233								
T.MR1	001566	2641#	13223											
T.MR2	001574	2644#	7887	8450										
T.MR3	001576	2645#												
T.SPAP	001562	2639#												
T.WC	001542	2631#	5413	5417	5467	5471	5602	5605	5608	5611	7104	7110	7122	7127
		7242	7248	7260	7265	10138	10141	10177	10220	10275	13236			
UBAERR=	000002	10196	12290#											
UFE =	000400	2017#	10442											
UFERR =	000004	3605	3653	10444	12281#									
UNLOAD=	000107	1981#												
UNS =	040000	2042#												
UNSERR=	040000	8424	12266#											
UPE =	020000	2022#	10458											
UPERR =	000100	5582	10460	12285#										
VV =	000100	2053#	3184											
WATSRT	035372	9735	9738	9741	9744	9747	9750	9753	9756	9759	9762	9765	9768	9771
		9774#												
WCE =	040000	2023#	10413											
WCERR =	000001	10140	12289#											
WCKERR=	000004	7079	7217	10415	12272#									
WLE =	004000	2039#												
WLERR =	004000	12263#												
WRODATA=	000123	1987#	4266	4347	4413	4466	4527	4586	4650	4715	4802	4877	4956	5025
		5102	5176	5238	5313	5382	5505	5636	5720	5806	5891	5976	6495	6582
		6672	6771	6970	7031	7167	7296	7391	7411	7457	7558	7626	7849	7962
		7981	8949											
WRDCNT	001706	2712#	10969	10997*	11045*	11169								
WRONUM	001704	2711#	10968	10998*	11044*	11168								
WRHEAD=	000127	1989#	4113	4238	4314	4393	4445	4499	4560	4621	6409	6742	6899	7357
		8379	8489	8533	11253									
WRL =	004000	2056#	3082											
WRTCHK=	000131	1990#	6989	7046	7066	7182	7204	7313	7685					
WRTGAT=	040000	2075#												

G05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 265
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0264

[illegible]

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 266
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0265

[illegible]

[illegible]

SEQ 0267

STPB 001152
STPFLG 001157

SEQ 0268

```
.SASTA= ***** U
.SX    = 000224
```

L05

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26MACY11 27(1006) 31-JAN-77 16:34 PAGE 271
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0269

COMMEN	1918#														
ENDCOM	1918#														
ERROR	1812#	2922	2951	3000	3034	3043	3068	3075	3099	3138	3146	3154	3159	3168	3173
	3179	3182	3187	3194	3221	3228	3231	3237	3240	3244	3261	3268	3270	3277	3283
	3286	3293	3311	3318	3321	3328	3339	3354	3362	3368	3375	3398	3406	3412	3426
	3431	3437	3444	3453	3459	3465	3471	3477	3482	3488	3494	3500	3513	3516	3522
	3528	3540	3543	3554	3560	3576	3583	3586	3600	3606	3609	3624	3631	3634	3648
	3654	3674	3681	3688	3696	3703	3710	3717	3721	3727	3740	3747	3756	3768	3771
	3782	3789	3797	3817	3824	3845	3853	3858	3863	3871	3882	3890	3906	3914	3919
	3924	3932	3938	3972	3984	3986	3997	4008	4018	4025	4034	4047	4090	4110	4126
	4129	4132	4151	4154	4157	4165	4172	4180	4183	4184	4185	4191	4200	4235	4258
	4261	4264	4275	4284	4294	4311	4338	4340	4344	4356	4365	4374	4390	4407	4410
	4422	4428	4442	4460	4463	4475	4481	4496	4521	4524	4536	4542	4557	4580	4583
	4594	4597	4615	4633	4636	4663	4666	4678	4681	4687	4695	4712	4729	4732	4752
	4755	4761	4768	4799	4815	4818	4834	4836	4842	4849	4875	4889	4892	4903	4905
	4911	4918	4949	4965	4968	4982	4985	4991	4999	5023	5037	5040	5051	5054	5060
	5068	5100	5115	5118	5130	5133	5140	5148	5173	5188	5191	5202	5205	5211	5219
	5235	5251	5254	5271	5274	5285	5293	5310	5325	5328	5343	5345	5356	5364	5379
	5391	5396	5405	5411	5418	5432	5444	5449	5458	5465	5472	5493	5577	5583	5592
	5599	5606	5612	5633	5649	5652	5668	5671	5683	5686	5692	5700	5717	5733	5736
	5756	5759	5771	5774	5780	5788	5803	5819	5822	5843	5846	5857	5863	5871	5888
	5904	5907	5924	5927	5939	5942	5948	5956	5973	5989	5993	6009	6012	6024	6027
	6033	6041	6060	6083	6086	6092	6100	6119	6140	6143	6149	6157	6176	6198	6201
	6207	6215	6233	6254	6257	6263	6271	6289	6310	6313	6319	6327	6342	6354	6357
	6369	6381	6387	6408	6421	6424	6436	6444	6445	6446	6453	6461	6484	6504	6507
	6519	6522	6528	6536	6580	6631	6635	6661	6681	6684	6696	6699	6705	6713	6739
	6757	6760	6780	6786	6795	6804	6810	6821	6824	6830	6838	6856	6862	6868	6876
	6896	6910	6913	6926	6940	6943	6967	6983	6986	6998	7001	7025	7040	7043	7055
	7058	7064	7075	7081	7102	7111	7120	7128	7161	7176	7179	7191	7194	7201	7213
	7219	7240	7249	7258	7266	7289	7305	7308	7322	7325	7369	7372	7389	7400	7403
	7414	7422	7427	7443	7446	7448	7455	7468	7472	7483	7494	7497	7505	7513	7553
	7572	7581	7586	7598	7602	7605	7608	7620	7623	7644	7648	7658	7662	7665	7667
	7679	7682	7701	7705	7720	7732	7735	7739	7741	7753	7756	7768	7780	7786	7788
	7792	7812	7824	7828	7831	7843	7846	7858	7864	7867	7882	7885	7891	7902	7917
	7932	7938	7955	7971	7974	8007	8013	8018	8036	8048	8051	8063	8081	8086	8090
	8095	8107	8109	8127	8139	8142	8145	8165	8168	8183	8195	8199	8211	8214	8227
	8230	8235	8238	8250	8253	8287	8293	8312	8324	8327	8331	8334	8346	8349	8361
	8364	8376	8404	8407	8421	8427	8430	8445	8448	8454	8461	8466	8480	8483	8498
	8501	8519	8527	8542	8545	8579	8593	8609	8613	8631	8648	8661	8677	8681	8697
	8738	8746	8748	8758	8761	8767	8776	8784	8892	8900	8904	8909	8912	8921	8924
	8928	8931	8942	8954	8957	8964	11217	11233	11236	11240	11243	11246	11268	11271	
ESCAPE	1918#														
GETPRI	1918#	9117													
GETSWR	1918#	2791#													
INITSS	2717#	4124	4149	4178	4711	4797	4873	4947	5021	5098	5171	5309	5378	5431	5492
	5632	5716	5802	5887	5972	6059	6118	6175	6232	6298	6341	6368	6406	6434	6483
	6579	6660	6738	6808	6894	6924	6966	7024	7063	7160	7200	7288	7387	7453	7481
	7552	7719	7787	7811	7845	7866	7890	7916	7954	8035	8088	8126	8144	8182	8311
	8429	8453	8460	8518	8526	8578	8647	8736	8890	11216	11245				
LOADLS	2719#	4714	4734	4825	4876	4973	5024	5120	5175	5312	5330	5503	5635	5658	5673
	5719	5742	5761	5805	5833	5847	5890	5914	5929	5975	5999	6014	6073	6130	6188
	6244	6300	6344	6371	6494	6509	6581	6671	6686	6741	6770	6811	6846	6927	6969
	6988	7030	7045	7065	7166	7181	7203	7295	7312	7390	7433	7456	7557	7588	7610
	7625	7649	7669	7684	7722	7743	7758	7770	7814	7833	7848	7870	7919	7961	7980
	7997	8038	8053	8065	8097	8129	8147	8185	8201	8216	8240	8276	8314	8336	8351
	8366	8378	8409	8432	8470	8488	8532	11223	11252						

M05

 RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
 DZRGKD.P11 31-JAN-77 16:26

 MACY11 27(1006) 31-JAN-77 16:34 PAGE 272
 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0270

MSG	2897#	2899	2928#	2930	2957#	2960	3211#	3213	3248#	3250	3300#	3302	3344#	3346	3377#
	3379	3563#	3565	3610#	3612	3658#	3660	3729#	3731	3772#	3774	3830#	3832	3896#	3898
	3942#	3944	4095#	4097	4215#	4217	4297#	4299	4377#	4379	4430#	4432	4482#	4484	4543#
	4545	4602#	4604	4699#	4701	4773#	4775	4854#	4856	4922#	4924	5003#	5005	5073#	5075
	5153#	5155	5224#	5226	5297#	5299	5369#	5371	5422#	5424	5476#	5478	5622#	5624	5705#
	5707	5792#	5794	5876#	5878	5961#	5963	6046#	6048	6105#	6107	6162#	6164	6219#	6221
	6275#	6277	6331#	6333	6358#	6360	6392#	6394	6472#	6474	6541#	6543	6647#	6649	6721#
	6723	6881#	6883	6956#	6958	7002#	7004	7138#	7140	7276#	7278	7329#	7331	7534#	7536
	7708#	7710	7798#	7801	7906#	7908	7940#	7943	8021#	8024	8113#	8115	8169#	8171	8294#
	8296	8503#	8505	8566#	8568	8634#	8636	8700#	8702	8864#	8866				
MULT	1918#														
NEWST	1918#	2897	2928	2958	3211	3248	3300	3344	3377	3563	3610	3658	3729	3772	3830
	3896	3942	4095	4215	4297	4377	4430	4482	4543	4602	4699	4773	4854	4922	5003
	5073	5153	5224	5297	5369	5422	5476	5622	5705	5792	5876	5961	6046	6105	6162
	6219	6275	6331	6358	6392	6472	6541	6647	6721	6881	6956	7002	7138	7276	7329
	7534	7708	7799	7906	7941	8022	8113	8169	8294	8503	8566	8634	8700	8864	
OPCHK	2721#	4731	4754	4817	4835	4891	4904	4967	4984	5039	5053	5117	5132	5190	5204
	5357	5344	5651	5670	5685	5735	5758	5773	5821	5845	5906	5926	5941	5992	6011
	6026	6085	6142	6200	6256	6312	6356	6423	6506	6521	6634	6683	6698	6759	6823
	6912	6942	6985	7000	7042	7057	7178	7193	7307	7324	7371	7402	7425	7445	7470
	7496	7584	7601	7607	7622	7646	7661	7666	7681	7703	7734	7740	7755	7830	7884
	7973	8016	8108	8141	8167	8213	8229	8237	8252	8326	8333	8348	8363	8406	8447
	8482	8500	8544	8611	8679	8747	8760	8910	8929	8956	11235	11242	11270		
POP	1918#	9372	9373	9826	9919	10116	10984	11003	11007	11171	11428	11817	11866	11898	11899
PUSH	1918#	9333	9335	9356	9815	9906	10090	10873	10884	11387	11791	11846	11879	11885	
REPORT	1918#														
RESDC	2718#	4762	4843	4912	4993	5062	5142	5213	5358	5694	5782	5865	5950	6035	6094
	6151	6209	6265	6321	6455	6530	6707	6832	6870	7507					
RKLOAD	2720#	4727	4750	4813	4832	4887	4901	4963	4980	5035	5049	5113	5128	5186	5200
	5323	5341	5389	5442	5647	5666	5681	5731	5754	5769	5817	5841	5855	5902	5922
	5937	5987	6007	6022	6081	6138	6196	6252	6308	6352	6379	6502	6517	6679	6694
	6755	6778	6819	6854	6908	6938	6981	6996	7038	7053	7073	7174	7189	7211	7303
	7320	7367	7398	7441	7492	7596	7618	7656	7677	7730	7751	7766	7778	7822	7841
	7856	7880	7969	8005	8046	8061	8105	8137	8193	8209	8225	8248	8285	8322	8344
	8359	8374	8419	8442	8478	8496	8540	8551	8659	8744	8756	8898	8907	8919	8926
	8940	11231	11266												
SCOPE	1813#	2904	2935	2997	3218	3258	3308	3351	3395	3573	3621	3671	3737	3779	3840
	3903	3964	4104	4231	4307	4386	4438	4492	4552	4611	4708	4783	4864	4932	5013
	5083	5163	5232	5307	5376	5429	5490	5630	5714	5800	5885	5970	6057	6116	6173
	6230	6286	6339	6366	6401	6480	6554	6656	6735	6889	6964	7020	7156	7286	7348
	7550	7717	7808	7914	7952	8032	8124	8180	8308	8515	8548	8573	8642	8715	8879
	9054														
SETPRI	1918#	11695													
SETTRA	11939#	11948	11949	11950	11951	11953	11955	11956	11957	11958	11959	11960	11961	11962	11963
	11964	11965	11966	11967	11968	11969	11970	11971	11972	11973	11974	11975	11976	11977	11978
	11979	11980													
SETUP	1918#	2741													
SKIP	1918#														
SLASH	1918#														
SPACE	1918#														
STARS	1918#	2176	2178	2185	2198	2273	2276	2897	2903	2928	2934	2958	2996	3211	3217
	3248	3257	3300	3307	3344	3350	3377	3394	3563	3572	3610	3620	3658	3670	3729
	3736	3772	3778	3830	3839	3896	3902	3942	3963	4095	4103	4215	4230	4297	4306
	4377	4385	4430	4437	4482	4491	4543	4551	4602	4610	4699	4707	4773	4782	4854
	4863	4922	4931	5003	5012	5073	5082	5153	5162	5224	5231	5297	5306	5369	5375
	5422	5428	5476	5489	5622	5629	5705	5713	5792	5799	5876	5884	5961	5969	6046

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
DZR6KD.P11 31-JAN-77 16:26

MACY11 27(1006) 31-JAN-77 16:34 PAGE 273
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0271

[illegible]

B06

RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC MACY11 27(1006) 31-JAN-77 16:34 PAGE 274
DZR6KD.P11 31-JAN-77 16:26 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0272

. ABS. 065414 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZR6KD.BIN,DSKZ:DZR6KD.SEQ/CRF/SOL/DOC=DZR6KD
RUN-TIME: 32 34 3 SECONDS
RUN-TIME RATIO: 387/71=5.4
CORE USED: 30K (60 PAGES)

DOCUMENT PAGES: 272

EOF1DZR6KDSEQ

00010000

770323

POP10 411