

RP11C/RP03

RELIABILITY DIAG
MD-11-DZRPB-E

EP-DZRPB-E-DL-A
COPYRIGHT © 72-77
FICHE 1 OF 1

AUG 1977
digital
MADE IN USA

B01

EOF1DZRYKESEQ

00010000

770712

PDP10 411

\$HOR1DZRPBESEQ

00010000

770712

CO1

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 1
DZRPBE.P11 29-APR-77 14:41

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRPB-E-D
PRODUCT NAME: RP11C RELIABILITY DIAGNOSTIC
DATE CREATED: MAY 1977
MAINTAINER: DIAGNOSTIC RELEASE ENGINEERING
AUTHOR: JOE STUBBLEBINE, WARREN MONCSKO

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED 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) 1972, 1977 BY DIGITAL EQUIPMENT CORPORATION

109 RP03. THE PROGRAM CONSIST OF SEVEN TESTS ANY ONE OF
110 WHICH IS SELECTABLE BY THE OPERATOR. A CONVERSATION
111 MODE EXISTS WHICH ALLOWS THE OPERATOR TO DEFINE TEST
112 PARAMETERS.
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD FAMILY PROCESSOR
RP11C DISK PACK CONTROLLER WITH UP TO EIGHT RP03
DRIVES
ASR33 OR EQUIVALENT.

2.2 STORAGE

BK OF STORAGE IS REQUIRED TO RUN THIS PROGRAM.

2.3 PRELIMINARY PROGRAMS

DZRPB RP11C DISKLESS DIAGNOSTIC

3.0 LOADING PRODEDURE

USE STANDARD PROCEDURE FOR ABS TAPES

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL SWITCHES DOWN FOR WORST CASE
TESTING-UNITO).

4.2 STARTING ADDRESS

THE PROGRAM MUST ALWAYS BE STARTED AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY USING ABS LOADER.
2. LOAD ADDRESS 200.
3. SET SWITCHES. ALL DOWN FOR WORST CASE-UNIT 0.
4. PRESS START.

THE PROGRAM NAME AND A SUMMARIZED SWITCH TABLE IS TYPED AFTER INITIAL START OF PROGRAM. IF USING MEMORY MANAGEMENT ON A SYSTEM WITH MORE THEN 8K OF MEMORY ALL OF THE LOADERS ARE MOVED FROM THE HIGHEST (28K MAXIMUM) CORE LOCATION TO INSIDE 8K. A RELOAD THE LOADERS STARTING ADDRESS IS GIVEN AFTER THE LOADERS ARE RELOCATED.

5. THE PROGRAM WILL LOOP AND TYPE PASS COUNT.
6. WHILE IN TEST 5 (DATA RELIABILITY) THE DISPLAY WILL CONTAIN THE NUMBER OF THE PATTERN CURRENTLY IN USE IN ORDER TO SHOW THE PROGRESS OF THE TEST.

5.0 OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 ALL SWITCHES DOWN IS WORST CASE TESTING FOR UNIT 0. PASS COUNT WILL BE TYPED OUT AT THE COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

SW<15>=1....HALT ON ERROR
 SW<14>=1....LOOP ON ERROR
 SW<13>=1....INHIBIT PRINTOUT
 SW<12>=1....INHIBIT BACKGROUND TEST
 SW<11>=1....RING BELL ON ERROR
 SW<10>=1....LOOP ON TEST
 SW<09>=1....INHIBIT DATA COMPARISON

SW<08>=1....ENTER CONVERSATION MODE.
 IF TEST 0-2 IS SELECTED USE ADDRESS
 CONVERSATION.
 IF TEST 3-7 IS SELECTED USE DATA
 CONVERSATION.

SW<07>=1....INHIBIT SEEKS BETWEEN WRITE AND READ
 DURING DATA TEST AND RANDOM TEST.

SW<06>=1....INHIBIT USE OF MEMORY MANAGEMENT

SW<05> USED TO CONTROL HOW MANY COMPARE ERRORS
 WILL BE TYPED OUT AS A RESULT OF A READ
 OPERATION IN THE DATA TEST.

SW<05>=1....CHECK FOR UP TO THREE COMPARE ERRORS
 WITHIN THE READ BUFFER AND TYPE ALL
 APPROPRIATE ERROR INFORMATION FOR EACH
 ERROR. NOTE IF THE DISK DISCOVERS AN
 ERROR, IT WILL FINISH READING THE
 CURRENT SECTOR AND THEN STOP. SO IF A
 VALID COMPARE ERROR IS ENCOUNTERED, AND
 THE PROGRAM CONTINUES SCANNING THE
 BUFFER, IT MAY GO BEYOND THE AREA WHERE
 THE DISK TRANSFERRED DATA. IF THIS
 HAPPENS, THE RECEIVED DATA WILL BE
 ZEROS.

SW<05>=0....CHECK FOR ONLY ONE COMPARE ERROR WITHIN
 THE READ BUFFER.

SW<04> USED TO CONTROL THE AMOUNT OF
 INFORMATION TYPED ON REREAD ATTEMPTS
 AFTER A READ FAILURE IN DATA TEST.

SW<04>=1....TYPE ALL ERROR INFORMATION ON
 REREAD ATTEMPT

SW<04>=0....TYPE THE ERROR INFORMATION ON THE FIRST
 READ ERROR ONLY. AFTER THE ERROR GOES
 AWAY OR IS UNRECOVERABLE, THE NUMBER OF
 REREADS IS THEN TYPED.

SW<03>=1....RUN TEST SELECTED BY SWITCH POSITIONS
 SW0 THRU SW2

SW<00>THRU SW<02>	TEST SELECTED
0	ADDRESS TEST 0
1	ADDRESS TEST 1
2	ADDRESS TEST 2
3	TEST3 - WRITE CHECK TEST
4	TEST4 - MEMORY ADDRESS TEST

HO1

5 TEST5 - DATA RELIABILITY
6 TEST6 - RANDOM TEST
7 TEST7 - POWER FAIL TEST

NOTE

IF IT IS DESIRED TO SELECT AN INDIVIDUAL
TEST, ALSO SET SW<10> LOOP ON TEST.

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED AT THE END OF EACH
SUBTEST AND PROVIDES THE ABILITY TO LOOP ON AN
ERROR. WHENEVER AN ERROR IS DETECTED, AN ERROR FLAG
IS SET. THIS FLAG IS TESTED BY THE SCOPE ROUTINE.
IF SET, AND LOOP ON ERROR SW<14> IS SET, THE PROGRAM
WILL LOOP BACK AND REPEAT THE CONDITIONS CAUSING THE
ERROR. PRIOR TO EACH SCOPE CALL THE LOOP ADDRESS IS
MOVED INTO LOCATION LAD. ONCE THE PROGRAM STARTS
LOOPING ON AN ERROR, IT WILL CONTINUE LOOPING EVEN
THOUGH THE ERROR MAY BE INTERMITTENT. TO GO OUT OF
THE LOOP RESET SW<14>.

5.2.2 HLT

THIS ROUTINE IS ENTERED UPON DETECTION OF AN ERROR.
IT WILL TYPE THE PC OF THE ERROR AND ADDITIONAL
ERROR INFORMATION. THIS ROUTINE TEST FOR HALT ON
ERROR, INHIBIT TYPEOUTS, AND RING THE BELL. IT ALSO
SETS THE ERROR FLAG USED BY THE SCOPE ROUTINE.

5.2.3 BACKGROUND TEST

THIS TEST IS ENTERED BY THE PROGRAM WHILE WAITTING
FOR AN INTERRUPT. IT DOES A SERIES OF NEGATE BYTE
AND ROTATE BYTE LEFT INSTRUCTIONS TO PROVIDE WORSE
CASE NPR TIMING, AND BUS NOISE AND IT WILL TIMEOUT
IF AN INTERRUPT FAILS TO OCCUR. THE BACKGROUND TEST
MAY BE INHIBITED BY SETTING SW<12> WHICH CAUSE THE
PROGRAM TO DO A WAIT INSTRUCTION.

5.2.4 TRAP CATCHER

277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332

A".+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR +2.

6.0 ERRORS

6.1 WHEN ERRORS ARE ENCOUNTERED, THE ADDRESS OF THE ERROR ALONG WITH THE CONTENTS OF RPDS, RPER, RPCS, RPCA, RPDA, AND SUCA ARE TYPED. BY REFERRING TO THE LISTING ADDITIONAL INFORMATION CAN BE FOUND REGARDING THE CAUSE OF THE ERROR IN THE COMMENTS. WHEN APPROPRIATE, ADDITIONAL INFORMATION IS TYPED OUT SUCH AS EXPECTED AND RECEIVED RESULTS OF AN OPERATION. ALL INFORMATION IS IN OCTAL.

ERROR MESSAGE FORMAT

1. PC= PC OF FAILURE
STATUS ERROR
RPDS= CONTENTS OF RPDS
RPER= CONTENTS OF RPER
RPCS= CONTENTS OF RPCS
RPCA= CONTENTS OF RPCA
RPDA= CONTENTS OF RPDA
SUCA= CONTENTS OF SUCA
CYLINDER= CYLINDER ADDRESS OF THE ERROR
HEAD= HEAD ADDRESS OF THE ERROR
SECTOR= SECTOR ADDRESS OF THE ERROR
READ COUNTER= A SPECIAL COUNTER USED IN DATA TEST AND RANDOM TEST. INDICATES THE NUMBER OF READS OF THAT BLOCK. 0 = FIRST READ. IF LOOPING ON ERROR THE FIRST READ COUNTER ONLY IS VALID.
2. PC= PC OF FAILURE
COMPARE ERROR
EXPECTED= DATA EXPECTED
RECEIVED= DATA RECEIVED
CYLINDER= CYLINDER ADDRESS OF THE ERROR
HEAD= HEAD ADDRESS OF THE

J01

389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444

SECTOR= ERROR
 WORD COUNT INTO SECTOR= SECTOR ADDRESS OF
 READ NO. THE ERROR DISTANCE INTO
 SECTOR. COUNT
 STARTS AT ONE.
 INDICATES WHICH READ
 ATTEMPT IS IN
 PROGRESS.

3. TOTAL REREADS ON ERROR= TOTAL READS BEFORE
 RECOVERY. TOTAL OF
 24 INDICATES ERROR
 WAS UNRECOVERABLE

7.0 RESTRICTIONS

TEST 7 (POWER FAIL) WILL BE EXECUTED ONLY IF
 SELECTED BY THE SWITCHES. IT WILL HALT AT
 COMPLETION.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE PASS COUNT WILL BE TYPED OUT AT THE END OF EACH
 PASS THRU THE PROGRAM. DUE TO THE TIME NECESSARY TO
 RUN ANY INDIVIDUAL TESTS, TESTS ARE NOT ITERATED.
 IF YOU WISH TO LOOP ON ANY PARTICULAR TEST, SELECT
 THE TEST IN SWITCH POSITIONS SW<00> THRU SW<02> AND
 SET SW<03> AND SW<10>. WHEN IN TEST 5 (DATA
 RELIABILITY) ITS PROGRESS CAN BE MONITORED BY
 LOOKING AT THE DISPLAY. IT WILL CONTAIN THE NUMBER
 OF THE PATTERN CURRENTLY IN USE.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500.

8.3 ERROR INFORMATION

IF IT IS DESIRED TO HAVE THE ERROR INFORMATION
 OUTPUTTED TO THE PUNCH INSTEAD OF THE TELETYPE
 CHANGE THE FOLLOWING THREE LOCATIONS.

LOCATION	FROM	TO
1304	177564	177554
1332	177566	177556
1336	177564	177554

8.4 NON-STANDARD VECTOR AND REGISTER ADDRESSES

IF IT IS DESIRED TO CHANGE THE VECTOR OR REGISTER ADDRESSES IT CAN BE ACCOMPLISHED BY CHANGING THE CONTENTS OF THE FOLLOWING LOCATIONS:

LOCATION	FROM	TO
VECTOR	254	NEW DISK INTERRUPT TRAP LOCATION
STATUS	256	NEW INTERRUPT LOCATION
RPCS	176714	NEW DISK CONTROL ADDRESS
RPCS1	176715	NEW UPPER BYTE OF CONTROL REGISTER
RPWC	176716	NEW WORD COUNT REGISTER ADDRESS
RPBA	176720	NEW CURRENT ADDRESS REGISTER ADDRESS
RPCA	176722	NEW CYLINDER ADDRESS REGISTER ADDRESS
RPDA	176724	NEW DISK ADDRESS REGISTER ADDRESS
RPDA1	176725	NEW TRACK ADDRESS REGISTER ADDRESS
RPER	176712	NEW ERROR REGISTER ADDRESS
RPDS	176710	NEW DEVICE STATUS REGISTER ADDRESS
SUCA	176734	NEW SELECTED UNIT CYLINDER ADDRESS REGISTER ADDRESS

9.0 PROGRAM DISCRIPTION

9.1 ADDRESS TEST 0

IN THIS TEST THE PROGRAM SEEKS FROM 0 TO N AND BACK TO 0. N STARTS AT ZERO THEN INCREMENTS TO 1 AND UP THRU 625. DONE IS TIMED OUT, SEEK UNDERWAY IS TESTED, UNIT READY IS TIMED OUT. ATTENTION INTERRUPT AND THE INTERRUPT FLAG ARE TESTED, AND THE CONTENTS OF SUCA ARE CHECKED.

501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556

9.2 ADDRESS TEST 1

WRITE 5000(OCTAL) WORDS IN THE TEN SECTORS ON EACH TRACK. THE FIRST WORD OF EACH SECTOR IS THE CYLINDER ADDRESS AND THE REMAINING WORDS CONTAIN THE HEAD AND SECTOR ADDRESS. AFTER EACH WRITE OPERATION, THE CONTENTS OF RPCA AND RPDA ARE EXAMINED TO SEE THAT THEY UPDATED CORRECTLY. AFTER WRITING THE ENTIRE PACK, THE DATA IS READ BACK TEN SECTORS AT A TIME AND VERIFIED. IF THE FIRST WORD OF A SECTOR DOES NOT COMPARE THE WRONG CYLINDER WAS PROBABLY SELECTED. AN ERROR ON THE FIRST WORD IS INDICATED BY TYPING "CYL" AFTER THE ERRING DATA. IF ANY OTHER WORD FAILS THE WRONG HEAD OR SECTOR WAS PROBABLY SELECTED. IN THIS CASE THE RIGHT HALF OF THE DATA TYPED EQUALS THE SECTOR ADDRESS AND THE LEFT HALF EQUALS THE TRACK ADDRESS.

9.3 ADDRESS TEST 2

WRITE THE FIRST WORD OF EACH CYLINDER WITH THE CYLINDER ADDRESS. THEN SEEK FROM 625 TO 0. THEN SEEK TO 624 AND BACK TO 1 UNTIL THE ADDRESSES CROSS AND GO BACK TO ZERO. AFTER EACH SEEK VERIFY THE POSITION OF THE HEAD BY READING THE FIRST WORD OF THE CYLINDER AND COMPARING IT WITH THE SELECTED CYLINDER.

9.4 WRITECHECK TEST

THIS TEST VERIFIES THE WRITE CHECK LOGIC AND THE ABILITY OF THE HARDWARE TO FILL THE REMAINDER OF A SECTOR WITH ZEROS WHEN A PARTIAL SECTOR IS WRITTEN. IN THE WRITE CHECK PORTION A FLOATING ONE AND A

FLOATING ZERO PATTERN ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC. THE PATTERN IS WRITTEN AND WRITE CHECKED EXPECTING NO ERRORS. THE BUFFER IS THEN CLEARED AND THE DATA IS WRITE CHECKED AGAIN. AN ERROR IS EXPECTED. AFTER WRITE CHECK IS TESTED, A SECTOR IS WRITTEN WITH ALL ONES AND THEN A TWO WORD WRITE IS PERFORMED. THE ENTIRE SECTOR IS READ AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND THE REMAINDER SHOULD BE ZERO.

9.5 MEMORY TEST

THIS TEST CONSISTS OF TWO SEGMENTS. THE FIRST SEGMENT TESTS THE ACCESSIBILITY OF MEMORY. IF MEMORY MANAGEMENT IS NOT AVAILABLE, OR MEMORY MANAGEMENT IS DESELECTED, OR THE SYSTEM ONLY CONTAINS BK WITH MEMORY MANAGEMENT THEN EACH LOCATION FROM THE END OF THE PROGRAM TO THE TOP OF MEMORY (NOT TO EXCEED 28K) IS WRITTEN WITH ITS ADDRESS. THIS DATA IS WRITTEN ON THE DISK. THE MEMORY IS CLEARED AND THE DATA IS READ BACK AND VERIFIED. IF MEMORY MANAGEMENT IS AVAILABLE AND SELECTED WITH MORE THAN BK OF CORE, UP TO A 20K BUFFER IS WRITTEN WITH ITS ADDRESS. THE BUFFER IS THEN WRITTEN ON THE DISK. THE BUFFER IS THEN CLEARED AND THE DISK IS READ AND CHECKED. THE BUFFER IS THEN MOVED IN MEMORY BY 1K AND THE PROCESS CONTINUES UNTIL ALL AVAILABLE CORE HAS BEEN WRITTEN AND CHECKED. IN SEGMENT TWO, THE EXTENDED ADDRESS BITS ARE TESTED IF MEMORY MANAGEMENT IS AVAILABLE.

9.6 DATA TEST

DATA TEST VERIFIES THE DATA RELIABILITY OF THE DISK. THE SEQUENCE IS WRITE THE PACK, WRITE CHECK, AND READ IT 3 TIMES. IF SW<07>=0 THEN A SEEK OF 128 OR GREATER CYLINDERS IS PERFORMED BETWEEN EACH WRITE AND READ. ERRORS THAT OCCUR WITH THE RANDOM SEEKS ENABLED DURING WRITE CHECK AS STATUS ERRORS OR DURING READ BUT ARE RECOVERABLE AFTER FIRST RETRY, OR ARE ELIMINATED BY DISABLING RANDOM SEEKS ARE PROBABLY DUE TO VIBRATING HEADS AFTER A SEEK. THIS SEQUENCE CONTINUES FOR THE 15 PATTERNS DEFINED BELOW. IF A DATA ERROR IS ENCOUNTERED DURING A READ OPERATION, THE OPERATION IS REPEATED 20 TIMES OR UNTIL THE ERROR GOES AWAY. AFTER THE TENTH TIME THE HEADS ARE HOMED AND REPOSITIONED. WITH EACH READ ERROR THE READ RETRY NUMBER IS TYPED OUT ALONG WITH THE ERROR INFORMATION. THIS WAY IT CAN BE

DETERMINED IF AN ERROR IS RECOVERABLE OR NON-RECOVERABLE. IF A READ STATUS ERROR OCCURS AND IT IS A SOFT ERROR (PARITY ERROR), THE DATA IS COMPARED TO PROVIDE ADDITIONAL INFORMATION. WHEN USING MEMORY MANAGEMENT WITH MORE THAN 8K OF MEMORY DURING THE READ THE BUFFER IS CONTINUALLY STEPPING THROUGH MEMORY AT 1K STEPS SO THAT ALL OF MEMORY IS CHECKED WITH THE DATA PATTERN.

NUMBER	DATA PATTERN	NUMBER	DATA PATTERN
0	163126	11	167356
1	052525	12	156735
2	125252	13	135673

613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668

3	031463	14	073567
4	007417	15	177777 - 000000
5	010421	16	RANDOM DATA
6	021042		
7	042104		
10	104210		

THE LENGTH OF EACH DATA TRANSFER IS DETERMINED BY THE SIZE OF MEMORY AND IS INDICATED BY A TYPEOUT AT THE BEGINNING OF THE PROGRAM, IF IN CONVERSATION MODE.

9.7 RANDOM TEST

IN THIS TEST RANDOM DATA OF 700(OCTAL) WORDS IS WRITTEN ON RANDOM SECTORS. IT IS THEN WRITE CHECKED AND READ 10 TIMES. ACCORDING TO SW<07> THERE COULD BE A RANDOM SEEK OF 128 OR GREATER CYLINDERS BETWEEN EACH WRITE AND READ. ERRORS OCCURING DURING READ BUT ARE RECOVERABLE AFTER THE FIRST RETRY OR ARE GREATLY REDUCED BY DISABLING RANDOM SEEKS ARE PROBABLY DUE TO VIBRATING HEADS AFTER A SEEK. THIS IS REPEATED 2500 TIMES. THE READ RECOVERY TECHNIQUE IS THE SAME AS DESCRIBED UNDER DATA TEST(9.6).

9.8 POWER FAIL TEST

TESTS THE ABILITY OF THE RP11C TO SENSE POWER FAILURE AND TO HOME THE HEADS. AS SOON AS THE OPERATOR IS REQUESTED TO TURN OFF POWER, THE PROGRAM WILL LOOP READING A SECTOR FROM THE DISK. AFTER POWER IS RESTORED, THE PROGRAM CHECKS THAT THE HEADS ARE ON CYLINDER ZERO AND THAT THE CONTENTS OF MEMORY ABOVE THE PROGRAM HAS NOT BEEN AFFECTED BY THE POWER FAILURE.

9.9 CONVERSATION MODE

THERE ARE TWO CONVERSATION MODES AVAILABLE SELECTED BY SW<08>. IF TEST NUMBER 0-2 IS SELECTED IN SW<00-02> THEN THE PROGRAM WILL SELECT THE ADDRESS CONVERSATION MODE. THIS WILL ALLOW THE OPERATOR TO SPECIFY TWO CYLINDER ADDRESSES WHICH WILL SETUP A CONTINUOUS SEEK OPERATIONS FROM ONE TO THE OTHER. A SUB TEST SELECTION DURING ADDRESS CONVERSATION IS FOR A HEAD ALIGNMENT ROUTINE. DURING THIS ROUTINE THE SWITCH REGISTER DEFINATIONS ARE CHANGED AS FOLLOWS:
SW<00-04> SELECTS THE HEAD TO ALIGN.

669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724

SW<08-10> SELECTS THE DRIVE TO ALIGN.
TOGGLING SW<07> WILL CAUSE THE HEAD AND DRIVE
SWITCHES TO BE REREAD AND SELECTION DONE AGAIN.
ANOTHER SUB TEST SELECTION DURING ADDRESS
CONVERSATION MODE IS FOR CONTINUOUS HOME SEEKS.
CAUTION MUST BE EXERCISED DURING THIS SUB TEST.
CONTINUOUS UNNECESSARY USE OF HOME SEEKS COULD BE
DSTRUCTIVE TO A DRIVE.
DATA TEST CONVERSATION MODE MAY BE ENTERED BY
SELECTING TEST NUMBER 3-7 IN SW<00-02>. IF SELECTED
A NUMBER OF QUESTIONS WILL BE ASKED TO DETERMINE
TEST PARAMETERS. ALL NUMBER RESPONSES SHOULD BE IN
OCTAL FOLLOWED BY A CARRIAGE RETURN.

THE CONVERSATION IS AS FOLLOWS:

DATA TEST ONLY?(Y OR N)

IF THE OPERATOR RESPONDS YES, THE PROGRAM ENTERS THE
DATA MODE ONLY, TEST 5 AND TEST 6.

MULTI DRIVE MODE?(Y OR N)

WITHIN THE MULTI DRIVE MODE, THE PROGRAM ALLOWS THE
OPERATOR TO EXERCISE ALL SYSTEM DRIVES WITHOUT
RESTARTING THE PROGRAM. A COMPLETE PASS IS MADE ON
DRIVE ZERO AND THE PROGRAM THEN GOES TO THE NEXT
DRIVE UNTIL ALL DRIVES ARE DONE. AT THIS TIME THE
PASSCOUNT IS UPDATED AND TYPED OUT. THE PROGRAM
CYCLES BACK TO UNIT ZERO AND CONTINUES. BEFORE
TESTING STARTS ON A UNIT, THE UNIT NUMBER IS TYPED
OUT.

IF THE OPERATOR RESPONDS YES - THE PROGRAM ASKS FOR
THE NUMBER OF DRIVES. IF THE OPERATOR RESPONDS
NO - THE PROGRAM ASKS FOR WHICH DRIVE TO EXERCISE.

NUMBER OF DRIVES 1 TO 10 (OCTAL)?

RESPOND WITH THE NUMBER OF DRIVES ON THE SYSTEM.

IF THE OPERATOR RESPONDS "NO" TO MULTI DRIVE MODE
THE FOLLOWING QUESTION IS ASKED.

WHICH DRIVE?

RESPOND WITH UNIT NUMBER OF DRIVE TO TEST

OPTIONAL WORD COUNT?(Y OR N)

PROVIDES THE OPPORTUNITY TO SPECIFY YOUR OWN WORD
COUNT WHICH MAY NOT EXCEED THE STANDARD WORD COUNT.
THE STANDARD WORD COUNT IS TYPED OUT AT THE START OF

THE PROGRAM. IF RESPONSE IS NO - THE NEXT QUESTION
IS SKIPPED.

LENGTH (1 TO STANDARD WORD COUNT)?

SPECIFY WORD COUNT IN OCTAL.

DO YOU WISH TO SELECT THE DISK TEST ADDR?(Y OR N)

THIS WILL ALLOW THE OPERATOR TO SELECT A SPECIFIC
AREA OF THE DISK FOR TESTING.

IF THE OPERATOR RESPONDS "YES" THE FOLLOWING
QUESTIONS WILL BE ASKED.

STARTING CYLINDER
STARTING HEAD
STARTING TRACK

RESPOND WITH THE DESIRED ADDRESS IN OCTAL

OPTIONAL DATA PATTERN NO.?

YOU HAVE THE OPTION OF SELECTING ANY INDIVIDUAL
PATTERN OR SELECTING ALL PATTERNS.

PATTERN NO.	PATTERN	PATTERN NO.	PATTERN
0	163126	10	404210
1	052525	11	167356
2	125252	12	156735
3	031463	13	135673
4	007417	14	073567
5	010421	15	177777 - 000000
6	021042	16	RANDOM PATTERN
7	042104	17	SELECTS ALL PATTERNS

WRITE? (Y OR N)

WRITE CHECK? (Y OR N)

READ? (Y OR N)

THESE QUESTIONS ALLOW YOU TO SELECT THE OPERATIONS
TO BE PERFORMED IN THE DATA TESTS.
.ENDR

002

```

771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826

      .LIST ME
      .NLIST MC,MD,CND
      .ABS
;COPYRIGHT 1972,1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS
;GENERAL REGISTER ASSIGNMENTS
      R0=%0
      R1=%1
      R2=%2
      R3=%3
      R4=%4
      R5=%5
      SP=%6
      PC=%7

;STATUS REGISTER (PSW) BIT ASSIGNMENTS
      C=1
      V=2
      Z=4
      N=10
      T=20
      PRI7=340
      PRI6=300
      PRI5=240
      PRI4=200
      PRI3=140
      PRI2=100
      PRI1=40
      ;C BIT
      ;V BIT
      ;Z BIT
      ;N BIT
      ;T BIT
      ;PRIORITY LEVEL 7
      ;PRIORITY LEVEL 6
      ;PRIORITY LEVEL 5
      ;PRIORITY LEVEL 4
      ;PRIORITY LEVEL 3
      ;PRIORITY LEVEL 2
      ;PRIORITY LEVEL 1

;VECTOR ADDRESSES
      ERRVEC=4
      RESVEC=10
      TBITVEC=14
      IOTVEC=20
      PFVEC=24
      EMTVEC=30
      TRAPVEC=34
      ;ERROR VECTOR
      ;RESERVED INST VECTOR
      ;T BIT VECTOR
      ;IOT TRAP VECTOR
      ;POWER FAIL VECTOR
      ;EMT VECTOR
      ;TRAP VECTOR

;REGISTER ADDRESSES
      PSW=177776
      TKS=177560
      TKB=177562
      TPS=177564
      TPB=177566
      SWR=177570
      DISPLAY=177570
      ;PROCESSOR STATUS REGISTER
      ;KEYBOARD CSR
      ;ADDR OF KEYBOARD BUFFER
      ;TELEPRINTER CSR
      ;TELEPRINTER BUFFER
      ;CONSOLE SWITCH REGISTER
      ;CONSOLE DISPLAY REGISTER

;INITIAL STACK POINTER
      STKPTR=500
      ;PROGRAM STACK POINTER

;BIT ASSIGNMENTS
      B15=100000
      B14=400000
  
```


E02

827	020000	813=20000
828	010000	812=10000
829	004000	811=4000
830	002000	810=2000
831	001000	89=1000
832	000400	88=400
833	000200	87=200
834	000100	86=100
835	000040	85=40
836	000020	84=20
837	000010	83=10
838	000004	82=4
839	000002	81=2
840	000001	80=1

; MEMORY MANAGEMENT REGISTER ASSIGNMENTS

844	177572	SRO=177572
845	172340	KIPAR0=172340
846	172342	KIPAR1=172342
847	172344	KIPAR2=172344
848	172346	KIPAR3=172346
849	172350	KIPAR4=172350
850	172352	KIPAR5=172352
851	172354	KIPAR6=172354
852	172356	KIPAR7=172356
853	172300	KIPDR0=172300
854	172302	KIPDR1=172302
855	172304	KIPDR2=172304
856	172306	KIPDR3=172306
857	172310	KIPDR4=172310
858	172312	KIPDR5=172312
859	172314	KIPDR6=172314
860	172316	KIPDR7=172316
861	000006	RW=6
862	000000	UP=00

; INSTRUCTION EQUATES

866	104400	HLT=TRAP
868	104000	SCOPE=EMT

; HLT IS A TRAP TO THE ERROR ROUTINE

; SCOPE IS AN EMT TRAP

; INDEX OF MACROS

871	:	.SCOPE
872	:	.SAVE
873	:	.REST
874	:	.ERROR
875	:	.PRINT
876	:	.DUMP
877	:	.RAND
878	:	.READ
879	:	.PACK

; INDEX OF CALLS

882

883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938

```

SCOPE
SAVE
REST
HLT
PRINT
DUMP
DUMPF
SDUMP
SDUMPF
RAND
READ
PACK

```

```

.LIST      ME
.=200
MOV        #START,PC      ;GO TO START OF TEST
.=1000
ICNT:      0              ;CONTAINS PASS COUNT
LAD:       0              ;PROGRAM TRACE
;SCOPE (EMT) SERVICE ROUTINE
;THIS ROUTINE WILL LOOP IF AN ERROR OCCURED AND
;LOOP ON ERROR SWITCH IS SET (BIT 14). IF LOOPING IS INDICATED
;THE CONTENTS OF "LAD" EQUAL THE LOOP ADDRESS. IN ORDER
;TO LOOP ON ERROR, BIT 14 OF THE SWITCH REGISTER MUST BE SET AND
;LOCATION "ERRFLG" MUST BE NEGATIVE INDICATING AN ERROR. ONCE THE
;LOOP IS INITIATED IT WILL CONTINUE UNTIL SWITCH 14 IS CLEARED.

SCOPE$: BIT      #B14,2#SWR      ;LOOP ON ERROR?
        BEQ      2$              ;BRANCH IF NO
        TST      ERRFLG          ;IS THERE AN ERROR?
        BNE      1$              ;BRANCH IF YES
2$:      CLR      ERRFLG          ;RESET ERROR CONDITION
        RTI                     ;EXIT
1$:      MOV      LAD,(SP)        ;MODIFY RETURN ADDRESS
        RTI                     ;EXIT
;ROUTINE TO SAVE REGISTERS ON THE STACK.
;CALLED BY SAVE MACRO
SAVE$:   MOV      (SP)+,1$        ;SAVE RETURN PC
        MOV      R5,-(SP)
        MOV      R4,-(SP)

```


939	001046	010346				MOV	R3,-(SP)	
940	001050	010246				MOV	R2,-(SP)	
941	001052	010146				MOV	R1,-(SP)	
942	001054	010046				MOV	R0,-(SP)	
943	001056	016707	000000			MOV	1\$,PC	;RETURN
944	001062	000000			1\$:	0		;CONTAINS RETURN ADDRESS
945								;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
946								;CALLED BY REST MACRO
947	001064	012667	000020		REST\$:	MOV	(SP)+,1\$	;SAVE RETURN PC
948	001070	012600				MOV	(SP)+,R0	
949	001072	012601				MOV	(SP)+,R1	
950	001074	012602				MOV	(SP)+,R2	
951	001076	012603				MOV	(SP)+,R3	
952	001100	012604				MOV	(SP)+,R4	
953	001102	012605				MOV	(SP)+,R5	
954	001104	016707	000000			MOV	1\$,PC	;RETURN
955	001110	000000			1\$:	0		;CONTAINS RETURN ADDR
956								;ERROR SERVICE ROUTINE CALLED BY HLT
957								;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
958								;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
959	001112	005737	177570		ERROR:	TST	2\$SWR	;HALT ON ERROR?
960	001116	100001				BPL	3\$	;BRANCH IF NO
961	001120	000000				HALT		
962	001122	032737	004000	177570	3\$:	BIT	#B11,2\$SWR	;RING THE BELL?
963	001130	001403				BEQ	1\$	;BRANCH IF NO
964	001132	004567	000144			JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
965	001136	001250				BELL		
966	001140	032737	020000	177570	1\$:	BIT	#B13,2\$SWR	;SKIP TYPEOUT?
967	001146	001022				BNE	2\$	;BRANCH IF YES
968	001150	004567	000110			JSR	R5,PRINT\$	;PRINT MESSAGE
969	001154	001252				ERRPC		
970	001156	011667	000062			MOV	(6),HLTADS	;GET ERROR PC+2
971	001162	162767	000002	000054		SUB	#2,HLTADS	;MODIFY
972	001170	117767	000050	000044		MOVB	2\$HLTADS,HLTCTS	;SAVE HLT ARGUMENT
973	001176	016767	000042	000356		MOV	HLTADS,TTY	
974	001204	004767	000134			JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
975	001210	004767	016150			JSR	PC,MSG	;GO TO USER ERROR ROUTINE
976	001214	005737	177570		2\$:	TST	2\$SWR	;HALT ON ERROR?
977	001220	100001				BPL	4\$	;BRANCH IF NO
978	001222	000000				HALT		
979	001224	052767	100000	000006	4\$:	BIS	#B15,ERRFLG	;SET ERROR FLAG
980	001232	005267	000010			INC	ERRORS	;UPDATE ERROR COUNTER
981	001236	000002				RTI		
982	001240	000000				ERRFLG:	0	
983	001242	000000				HLTCTS:	0	
984	001244	000000				HLTADS:	0	;PC OF ERROR
985	001246	000000				ERRORS:	0	;ERROR COUNT
986	001250	000007				BELL:	.ASCIZ <7>	
987	001252	005015	005015	041520		ERRPC:	.ASCIZ <15><12><15><12>'PC= '	
988	001260	020075	000					
989		001264						
990								
991								
992	001264	032737	020000	177570		PRINTS:	BIT	#B13,2\$SWR
993	001272	001403				BEQ	PRNTF\$	;INHIBIT TYPEOUTS?
994	001274	062705	000002			ADD	#2,R5	;BRANCH IF NO
								;UPDATE RETURN ADDR

995	001300	000205			RTS	R5	
996	001302	105737	177564		PRNTFS: TSTB	2#TPS	;WAIT FOR PRINTER TO FINISH
997	001306	100375			BPL	-4	
998	001310	010546			MOV	R5, -(SP)	
999	001312	062716	000002		ADD	2, (SP)	;ADJUST RETURN PC
1000	001316	011505			MOV	(R5), R5	;GET MESSAGE ADDR
1001	001320	105715		1\$:	TSTB	(R5)	;CHECK FOR TERMINATOR
1002	001322	001002			BNE	2\$	
1003	001324	012605			MOV	(SP)+, R5	;GET RETURN ADDR
1004	001326	000205			RTS	R5	;RETURN
1005	001330	112537	177566	2\$:	MOVB	(R5)+, 2#TPB	;PRINT CHARACTER
1006	001334	105737	177564		TSTB	2#TPS	;WAIT TILL DONE
1007	001340	100375			BPL	-4	
1008	001342	000766			BR	1\$	
1009					:THIS ROUTINE TYPES A LOCATION IN OCTAL		
1010	001344	032737	020000	177570	PRINTR: BIT	#B13, 2#SWR	;INHIBIT TYPEOUT?
1011	001352	001406			BEQ	PRINTA	;BRANCH IF NO
1012	001354	000207			RTS	PC	
1013	001356	032737	020000	177570	PRINTS: BIT	#B13, 2#SWR	;INHIBIT TYPEOUT?
1014	001364	001405			BEQ	PRINTB	;BRANCH IF NO
1015	001366	000207			RTS	PC	
1016	001370	112767	000001	000140	PRINTA: MOVB	1, .PR	;SET ZERO FILL SWITCH
1017	001376	000402			BR	+.6	;SKIP
1018	001400	005067	000132		PRINTB: CLR	.PR	;SUPPRESS LEADING ZEROS
1019	001404	112767	177772	000125	MOVB	-6, .PR+1	;SET COUNT
1020	001412	010446			.PTIT: MOV	R4, -(SP)	;SAVE R4
1021	001414	012704	001540		MOV	1, .PR+2, R4	;SET POINTER TO FIRST CHARACTER
1022	001420	105014			CLRB	(R4)	;CLEAR FIRST BYTE
1023	001422	000413			BR	.PRF	;ROTATE FIRST BIT
1024	001424	105014			.PRL: CLRB	(R4)	;CLEAR BYTE OF CHAR
1025	001426	032767	000100	000102	BIT	100, .PR	;BIT TYPING MODE
1026	001434	001006			BNE	.PRF	;YES SKIP 2 ROTATES
1027	001436	006167	000120		ROL	TTY	;ROTATE BIT INTO C
1028	001442	106114			ROLB	(4)	;PACK IT
1029	001444	006167	000112		ROL	TTY	
1030	001450	106114			ROLB	(4)	
1031	001452	006167	000104		.PRF: ROL	TTY	
1032	001456	106114			ROLB	(4)	
1033	001460	105714			TSTB	(4)	;IS IT ZERO
1034	001462	001402			BEQ	+.6	;SKIP INC
1035	001464	105267	000046		INCB	.PR	;SET FILL SWITCH
1036	001470	105767	000042		TSTB	.PR	;CHECK FILL SWITCH
1037	001474	001402			BEQ	+.6	;SKIP BITSET
1038	001476	152724	000060		BISB	0, (4)+	;MAKE INTO ASCII CHAR
1039	001502	105267	000031		INCB	.PR+1	;INC COUNT
1040	001506	001346			BNE	.PRL	;REPEAT
1041	001510	022704	001540		CMP	1, .PR+2, R4	;EMPTY BUFFER
1042	001514	001002			BNE	+.6	;SKIP IF NOT
1043	001516	112724	000060		MOVB	0, (4)+	;LOAD ONE ZERO
1044	001522	105014			CLRB	(4)	;NULL TERMINATOR
1045	001524	004567	177534		JSR	R5, PRINTS	;PRINT MESSAGE
1046	001530	001540			.PR+2		
1047	001532	012604			MOV	(SP)+, R4	;RESTORE R4
1048	001534	000207			RTS	PC	
1049	001536	000012			.PR:	BLKW	12
1050	001562	000000			TTY:	0	

1051	001564			RAND\$:	JSR	PC,SAVE\$	;SAVE THE REGISTERS
1052	001564	004767	177246		MOV	LONUM,R0	;SET R0 WITH LOW
1053	001570	016700	000106		MOV	HINUM,R1	;SET R1 WITH HIGH
1054	001574	016701	000100		MOV	#-7,R3	;SET SHIFT COUNT
1055	001600	012703	177771		CLR	R2	
1056	001604	005002		1\$:	ASL	R0	;SHIFT R0 LEFT AND
1057	001606	006300			ROL	R1	;ROTATE CARRY INTO R1 AND
1058	001610	006101			ROL	R2	;ROTATE CARRY INTO R2
1059	001612	006102			INC	R3	;CHECK FOR DONE
1060	001614	005203			BNE	1\$	
1061	001616	001373			ADD	LONUM,R2	;ADD # TO MAKE X 129
1062	001620	066702	000056		ADC	R1	;PROPOGATE CARRY
1063	001624	005501			ADD	HINUM,R1	;ADD # TO MAKE X 129
1064	001626	066701	000046		ADC	R2	;PROPOGATE CARRY
1065	001632	005502			ADD	#1057,R0	
1066	001634	062700	001057		ADC	R1	;PROPOGATE CARRY
1067	001640	005501			ADC	R2	;PROPOGATE CARRY
1068	001642	005502			ADD	#47401,R1	
1069	001644	062701	047401		ADC	R2	
1070	001650	005502			ADD	#6,R2	
1071	001652	062702	000006		ADD	R2,R0	
1072	001656	060200			ADC	R1	
1073	001660	005501			MOV	R0,LONUM	
1074	001662	010067	000014		MOV	R1,HINUM	
1075	001666	010167	000006		JSR	PC,REST\$	;RESTORE THE REGISTERS
1076	001672	004767	177166		RTS	PC	
1077	001676	000207					
1078							
1079	001700	000000		HINUM:	0		
1080	001702	000000		LONUM:	0		
1081	001704	010346		READ\$:	MOV	R3,-(6)	;SAVE R3
1082	001706	012703	002014	1\$:	MOV	#INPUT\$,R3	;GET BUFFER ADDR
1083	001712	022703	002034	2\$:	CMP	#INPUT\$+20,R3	;BUFFER FULL?
1084	001716	001412			BEQ	4\$	;YES..TYPE ?
1085	001720	105737	177560		TSTB	#177560	;WAIT FOR A CHAR
1086	001724	100375			BPL	-4	
1087	001726	113713	177562		MOVB	#177562,(3)	;GET CHAR
1088	001732	142713	000200		BICB	#200,(3)	;GET RID OF JUNK
1089	001736	122713	000177		CMPB	#177,(3)	;IS IT A RUBOUT?
1090	001742	001004			BNE	3\$	;SKIP IF NO
1091	001744			4\$:			
1092	001744	004567	177314		JSR	R5,PRINT\$	;PRINT MESSAGE
1093	001750	002054			READM\$	1\$	
1094	001752	000755			BR	1\$	;CLEAR BUFFER AND START OVER
1095	001754	013737	177562	177566 3\$:	MOV	#TKB,#TPB	;ECHO THE CHAR
1096	001762	105737	177564		TSTB	#TPS	
1097	001766	100375			BPL	-4	;WAIT FOR READY
1098	001770	122723	000015		CMPB	#15,(3)+	;CHECK FOR RETURN
1099	001774	001346			BNE	2\$	;LOOP IF NOT RETURN
1100	001776	105063	177777		CLRB	-1(3)	;REMOVE THE RETURN
1101	002002	004567	177256		JSR	R5,PRINT\$	;PRINT MESSAGE
1102	002006	002060			READL\$		
1103	002010	012603			MOV	(6)+,R3	;RESTORE R3
1104	002012	000207			RTS	PC	;RETURN
1105							
1106	002014	000020		INPUT\$:	.BLKW	20	

1107	002054	006477	000012	READMS: .ASCIZ '??' (15) (12)
1108	002060	000012		READLS: .ASCIZ (12)
1109				
1110				: TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
1111				: PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
1112				
1113	002062			PACKS:
1114	002062	004767	176750	JSR PC, SAVES ; SAVE THE REGISTERS
1115	002066	005067	000242	CLR NUMS
1116	002072	005000		CLR RO
1117	002074	105760	002014	2\$: TSTB INPUTS(RO)
1118	002100	001402		BEQ 1\$
1119	002102	005200		INC RO
1120	002104	000773		BR 2\$
1121	002106	005300		1\$: DEC RO
1122	002110	004767	000166	JSR PC, PACS ; GET OCTAL CHAR
1123	002114	016767	000212 000212	MOV PK\$, NUMS ; PACK FIRST CHAR
1124	002122	004767	000154	JSR PC, PACS ; GET OCTAL CHAR
1125	002126	000241		CLC
1126	002130	006167	000176	ROL PK\$
1127	002134	006167	000172	ROL PK\$
1128	002140	006167	000166	ROL PK\$
1129	002144	056767	000162 000162	BIS PK\$, NUMS ; PACK SECOND CHAR
1130	002152	004767	000124	JSR PC, PACS ; GET OCTAL CHAR
1131	002156	000241		CLC
1132	002160	000367	000146	SWAB PK\$
1133	002164	006067	000142	ROR PK\$
1134	002170	006067	000136	ROR PK\$
1135	002174	056767	000132 000132	BIS PK\$, NUMS ; PACK THIRD CHAR
1136	002202	004767	000074	JSR PC, PACS ; GET OCTAL CHAR
1137	002206	000367	000120	SWAB PK\$
1138	002212	000241		CLC
1139	002214	006167	000112	ROL PK\$
1140	002220	056767	000106 000106	BIS PK\$, NUMS ; PACK FOURTH CHAR
1141	002226	004767	000050	JSR PC, PACS ; GET OCTAL CHAR
1142	002232	000367	000074	SWAB PK\$
1143	002236	000241		CLC
1144	002240	006167	000066	ROL PK\$
1145	002244	006167	000062	ROL PK\$
1146	002250	006167	000056	ROL PK\$
1147	002254	006167	000052	ROL PK\$
1148	002260	056767	000046 000046	BIS PK\$, NUMS ; PACK FIFTH CHAR
1149	002266	000402		BR PKEX1\$
1150	002270	062706	000002	PKEX\$: ADD #2, SP ; MODIFY STACK
1151	002274			PKEX1\$:
1152	002274	004767	176564	JSR PC, RESTS ; RESTORE THE REGISTERS
1153	002300	000207		RTS PC ; EXIT
1154				
1155	002302	005700		PACS: TST RO
1156	002304	100771		BMI PKEX\$
1157	002306	005067	000020	CLR PK\$
1158	002312	116067	002014 000012	MOVB INPUTS(RO), PK\$; GET INPUT CHAR
1159	002320	005300		DEC RO
1160	002322	042767	177770 000002	BIC #177770, PK\$; CLEAR UNWANTED BITS
1161	002330	000207		RTS PC
1162				

K02

RELIC RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 22
DZRPBE.PYI 29-APR-77 14:41

1163 002332 000000
1164 002334 000000
1165

PKS: 0
NUMS: 0

.TITLE RP11C RELIABILITY TEST

```

1166
1167
1168
1169
1170 002336 000005          START: RESET          ;CLEAR THE WORLD
1171 002340 012706 000500      MOV      #STKPTR,SP      ;SETUP STACK
1172 002344 004767 014340      JSR      PC,INIT        ;INITIALIZE VECTORS
1173 002350 004567 176726      JSR      RS,PRNTFS      ;FORCE PRINT THE MESSAGE
1174 002354 025760              HEADER
1175 002356 005767 023376      TST      HEADER          ;WAS RPO3(S) SIZED ALREADY?
1176 002362 001402              BEQ      RESTRT          ;IF YES, THEN DO NOT SIZE
1177 002364 004767 025140      JSR      PC,SUBSIZ       ;OTHERWISE, SIZE FOR RPO3(S).
1178 002370 005067 023270      RESTRT: CLR      FLAG      ;CLEAR PROGRAM FLAG
1179 002374 032737 000100 177570 BIT      #B6,#SWR      ;WANT TO USE MEMORY MANAGEMENT?
1180 002402 001034              BNE      2$              ;BR IF NO
1181 002404 012737 002474 000004 MOV      #2$,#ERRVEC      ;SETUP TRAP TEST FOR MEMORY MANAGEMENT
1182 002412 012737 000340 000006 MOV      #340,#ERRVEC+2
1183 002420 005737 177572      TST      #SR0              ;MEMORY MANAGEMENT?
1184 002424 005037 172340      CLR      #KIPAR0          ;YES! SET UP TO USE MEMORY MANAGEMENT
1185 002430 012737 000200 172342 MOV      #200,#KIPAR1      ;SECOND 4K PAGE
1186 002436 012737 007600 172356 MOV      #7600,#KIPAR7     ;I/O PAGE
1187 002444 012737 177406 172300 MOV      #400*256.-400+UP+RW,#KIPDR0 ;SET KIPDR0=RW UP 400 BLOCKS
1188 002452 012737 177406 172302 MOV      #400*256.-400+UP+RW,#KIPDR1 ;SET KIPDR1=RW UP 400 BLOCKS
1189 002460 012737 177406 172316 MOV      #400*256.-400+UP+RW,#KIPDR7 ;SET KIPDR7=RW UP 400 BLOCKS
1190 002466 005237 177572      INC      #SR0              ;TURN ON MEMORY MANAGEMENT
1191 002472 000403              BR      7$
1192 002474 052767 000100 023162 2$: BIS      #B6,FLAG      ;BIT6 SET = NO MEMORY MANAGEMENT
1193 002502 012737 000006 000004 7$: MOV      #ERRVEC+2,#ERRVEC      ;RESTORE TRAP CATCHER
1194 002510 005037 000006      CLR      #ERRVEC+2
1195 002514 005067 023240      CLR      HEADER          ;ASCII TERMINATOR SO ON RESTART THERE IS NO HEADER
1196 002520 004567 016242      JSR      RS,EXTMEN       ;SET UP DATA BUFFERS
1197 002524 005067 023222      CLR      SEEK1           ;INITALIZE SEEK RANDOM NUMBER GENERATOR
1198 002530 005067 176244      CLR      ICNT            ;CLEAR THE PASS COUNTER
1199 002534 005067 023154      CLR      DSKNOR          ;CLEAR UNIT FLAG
1200 002540 005067 023132      CLR      CYLINDER        ;CLEAR THE CYLINDER ADDRESS
1201 002544 005067 023130      CLR      DMA              ;CLEAR DAR REGISTERS
1202 002550 005067 023130      CLR      PATNU           ;CLEAR PATTERN COUNT
1203 002554 005067 023174      CLR      CNTA            ;CLEAR READ COUNTER FOR DATA AND RANDOM TEST
1204 002560 032737 000400 177570 BIT      #B8,#SWR      ;USE CONVERSATION MODE?
1205 002566 001005              BNE      LCONM           ;BRANCH IF YES
1206 002570 052767 070000 023066 BIS      #70000,FLAG
1207 002576 000167 000756      JMP      ADTST
1208
1209 002602 013746 177570      ;ENTER OPERATOR CONVERSATION MODE
1210 002606 042716 177770      LCONM: MOV      #SWR,-(SP)      ;PUT SWR ONTO STACK
1211 002612 022726 000003      BIC      #-10,(SP)        ;CLEAN OUT UNWANTED SWITCHES
1212 002616 101402              CMP      #3,(SP)+        ;TEST 3?
1213 002620 000167 017672      BLOS      1$              ;BR IF YES
1214 002624              JMP      CYLSK                  ;GO DO ADDRESS CONVERSATION
1215 002624 004567 176452      1$: JSR      RS,PRNTFS      ;FORCE PRINT THE MESSAGE
1216 002630 024532              SPECMES      ;STANDARD WORDS TRANSFERED =
1217 002632 016767 023052 176722 MOV      SWRDCT,TTY
1218 002640 004767 176534      JSR      PC,PRINTB      ;FORCE TYPE LOCATION - SUPPRESS ZEROS
1219 002644 004567 176432      JSR      RS,PRNTFS      ;FORCE PRINT THE MESSAGE
1220 002650 024561              CON1          ;ASK ABOUT DATA TEST ONLY
1221 002652 004767 177026      JSR      PC,READ$      ;INPUT MESSAGE

```


1222	002656	122767	000131	177130	CMPB	#131, INPUTS	; TEST FOR YES
1223	002664	001003			BNE	+10	; BRANCH IF NO
1224	002666	052767	002000	022770	BIS	#B10, FLAG	; SET DATA TEST ONLY FLAG
1225	002674	004567	176402		JSR	R5, PRNTRF\$	; FORCE PRINT THE MESSAGE
1226	002700	024614			CON2		; ASK ABOUT MULTI DRIVE MODE
1227	002702	004767	176776		JSR	PC, READ\$	; INPUT MESSAGE
1228	002706	122767	000131	177100	CMPB	#131, INPUTS	; TEST FOR YES
1229	002714	001040			BNE	DATTES	; BRANCH IF NO
1230	002716	052767	004000	022740	BIS	#B11, FLAG	; SET MULTI UNIT FLAG
1231	002724						
1232	002724	004567	176352		JSR	R5, PRNTRF\$	; FORCE PRINT THE MESSAGE
1233	002730	024650			CON3		; GET NO. OF UNITS
1234	002732	004767	176746		JSR	PC, READ\$	; INPUT MESSAGE
1235	002736	004767	177120		JSR	PC, PACK\$	; CONVERT INPUT TO A NUMBER
1236	002742	005767	177366		TST	NUM\$	; IS IT ZERO
1237	002746	001766			BEQ	DSKDR	
1238	002750	162767	000001	177356	SUB	#1, NUM\$	
1239	002756	022767	000010	177350	CMP	#10, NUM\$	; IS NO. TOO HIGH
1240	002764	101757			BLOS	DSKDR	
1241	002766	016767	177342	022720	MOV	NUM\$, DSKNOR	; SAVE HIGHEST UNIT NO.
1242	002774	042767	177770	022712	BIC	#177770, DSKNOR	
1243	003002	000241			CLC		
1244	003004	006167	022704		ROL	DSKNOR	
1245	003010	006167	022700		ROL	DSKNOR	
1246	003014	000423			BR	ASKWC	
1247	003016						
1248	003016	004567	176260		JSR	R5, PRNTRF\$	; FORCE PRINT THE MESSAGE
1249	003022	024712			CON4		; ASK UNIT NUMBER
1250	003024	004767	176654		JSR	PC, READ\$	; INPUT MESSAGE
1251	003030	004767	177026		JSR	PC, PACK\$	; CONVERT INPUT TO A NUMBER
1252	003034	022767	000010	177272	CMP	#10, NUM\$	; IS NO = OR > 10
1253	003042	101765			BLOS	DATTES	; NO
1254	003044	000241			CLC		
1255	003046	006167	177262		ROL	NUM\$	
1256	003052	006167	177256		ROL	NUM\$	
1257	003056	056767	177252	022600	BIS	NUM\$, FLAG	; SAVE UNIT UNDER TEST
1258	003064						
1259	003064	004567	176212		JSR	R5, PRNTRF\$	; FORCE PRINT THE MESSAGE
1260	003070	024731			CON5		; ASK ABOUT OPTIONAL WORD COUNT
1261	003072	004767	176606		JSR	PC, READ\$	; INPUT MESSAGE
1262	003076	122767	000131	176710	CMPB	#131, INPUTS	; TEST FOR YES
1263	003104	001034			BNE	TKSR	; ASK ABOUT OPTIONAL DAR
1264	003106						
1265	003106	004567	176170		JSR	R5, PRNTRF\$	; FORCE PRINT THE MESSAGE
1266	003112	024771			CON6		; ASK LENGTH OF WC
1267	003114	004767	176564		JSR	PC, READ\$	; INPUT MESSAGE
1268	003120	004767	176736		JSR	PC, PACK\$	; CONVERT INPUT TO A NUMBER
1269	003124	005767	177204		TST	NUM\$	
1270	003130	001766			BEQ	WCCON	
1271	003132	016767	022552	022574	MOV	SWRDCT, WORK	
1272	003140	005267	022570		INC	WORK	
1273	003144	026767	022564	177162	CMP	WORK, NUM\$	; IS NO. GREATER THAN AVAILABLE CORE?
1274	003152	101755			BLOS	WCCON	; YES ASK FOR COUNT AGAIN
1275	003154	016767	177154	022526	MOV	NUM\$, SWRDCT	; OPERATING WORD COUNT
1276	003162	016767	022522	022504	MOV	SWRDCT, WRDCT	
1277	003170	052767	000002	022466	BIS	#B1, FLAG	; OPERATOR SELECTED WORD COUNT

1278	003176				TKSR:			
1279	003176	004567	176100			JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1280	003202	025042				CON7A		;ASK ABOUT DISK ADDR
1281	003204	004767	176474			JSR	PC,READ\$	;INPUT MESSAGE
1282	003210	122767	000131	176576		CMPB	#131,INPUT\$	;WILL OPERATOR SUPPLY ADDR?
1283	003216	001055				BNE	OPPAT	;BRANCH IF NO
1284	003220	052767	000040	022436		BIS	#B5,FLAG	
1285	003226				OPDAR:			
1286	003226	004567	176050			JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1287	003232	025147				CON7C		;GET CYLINDER ADDR
1288	003234	004767	176444			JSR	PC,READ\$	;INPUT MESSAGE
1289	003240	004767	176616			JSR	PC,PACK\$	;CONVERT INPUT TO A NUMBER
1290	003244	022767	000626	177062		CMP	#626,NUM\$	;IS CYLINDER LEGAL
1291	003252	101765				BLOS	OPDAR	
1292	003254	016767	177054	022404		MOV	NUM\$,SCYL	;SAVE ADDR
1293	003262				OPDA1:			
1294	003262	004567	176014			JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1295	003266	025021				CON7		;GET HEAD ADDR
1296	003270	004767	176410			JSR	PC,READ\$	;INPUT MESSAGE
1297	003274	004767	176562			JSR	PC,PACK\$	;CONVERT INPUT TO A NUMBER
1298	003300	022767	000024	177026		CMP	#24,NUM\$	
1299	003306	101765				BLOS	OPDA1	;BRANCH IF HEAD ADDR TOO HIGH
1300	003310	016767	177020	022352		MOV	NUM\$,SHED	;SAVE ADDR
1301	003316				OPDA2:			
1302	003316	004567	175760			JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1303	003322	025126				CON7B		;GET SECTOR ADDR
1304	003324	004767	176354			JSR	PC,READ\$	;INPUT MESSAGE
1305	003330	004767	176526			JSR	PC,PACK\$	;CONVERT INPUT TO A NUMBER
1306	003334	022767	000012	176772		CMP	#12,NUM\$	;IS SECTOR ADDR TOO HIGH?
1307	003342	101765				BLOS	OPDA2	
1308	003344	016767	176764	022320		MOV	NUM\$,SSEC	;SAVE ADDR
1309					OPPAT:			
1310	003352					JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1311	003352	004567	175724			CON8		;ASK ABOUT DATA PATTERNS
1312	003356	025167				JSR	PC,READ\$	;INPUT MESSAGE
1313	003360	004767	176320			JSR	PC,PACK\$	;CONVERT INPUT TO A NUMBER
1314	003364	004767	176472			CMP	#20,NUM\$	;TEST FOR CORRECT NO
1315	003370	022767	000020	176736		BLOS	OPPAT	;ASK AGAIN
1316	003376	101765				CMP	#17,NUM\$	
1317	003400	022767	000017	176726		BEQ	OPWRT	;DATA PATTERN UNDER PROGRAM CONTROL
1318	003406	001411				BIS	#B15,FLAG	;SET PROGRAM FLAG
1319	003410	052767	100000	022246		MOV	NUM\$,PATNU	;OPERATOR WANTS TO SELECT DATA
1320	003416	016767	176712	022260		CLC		
1321	003424	000241				ROL	PATNU	
1322	003426	006167	022252			BIC	#70000,FLAG	;CLEAR OP MODE BITS IN FLAG
1323	003432	042767	070000	022224	OPWRT:	JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1324	003440	004567	175636			CON9		;ASK ABOUT WRITE
1325	003444	025215				JSR	PC,READ\$	;INPUT MESSAGE
1326	003446	004767	176232			CMPB	#131,INPUT\$	;TEST FOR YES
1327	003452	122767	000131	176334		BNE	OPRD	;ASK ABOUT WRITE CHECK
1328	003460	001003				BIS	#B14,FLAG	;YES SET FLAG BIT
1329	003462	052767	040000	022174	OPRD:			
1330	003470					JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
1331	003470	004567	175606			CON11		;ASK ABOUT READ
1332	003474	025265				JSR	PC,READ\$	;INPUT MESSAGE
1333	003476	004767	176202					

Line	Address	Offset	Label	Operation	Comments
1334	003502	122767	000131	176304	CMPB #131, INPUTS ; TEST FOR YES ANSWER
1335	003510	001003			BNE OPWCK
1336	003512	052767	010000	022144	BIS #B12, FLAG ; SET FLAG TO READ
1337	003520				OPWCK:
1338	003520	004567	175556		JSR R5, PRNTPS ; FORCE PRINT THE MESSAGE
1339	003524	025236			CON10 ; ASK ABOUT WRITE CHECK
1340	003526	004767	176152		JSR PC, READS ; INPUT MESSAGE
1341	003532	122767	000131	176254	CMPB #131, INPUTS
1342	003540	001003			BNE CHKMOD
1343	003542	052767	020000	022114	BIS #B13, FLAG ; SET WRITE CHECK FLAG
1344	003550	032767	070000	022106	CHKMOD: BIT #70000, FLAG ; MAKE SURE SOME OPERATION WAS SELECTED
1345	003556	001725			BEQ OPWRT
1346					
1347					
1348	003560	005737	000042		ADTST: TST #42 ; UNDER MONITOR CONTROL?
1349	003564	001444			BEQ 1\$; BRANCH IF NO
1350	003566	005067	022122		CLR DSKNOR
1351	003572	012777	000001	022040	3\$: MOV #1, ARPCS ; CLEAR THE RP11C
1352	003600	116777	022110	022034	MOVB DSKNOR, ARPCS1 ; SELECT THE DRIVE
1353	003606	005777	022046		TST ARPDS ; IS THE UNIT READY?
1354	003612	100003			BPL 2\$; BRANCH IF NO
1355	003614	005267	022074		INC DSKNOR ; UPDATE UNIT NUMBER
1356	003620	000764			BR 3\$
1357	003622	005367	022066		2\$: DEC DSKNOR ; DSKNOR = NUMBER OF UNITS
1358	003626	000241			CLC
1359	003630	006167	022060		ROL DSKNOR
1360	003634	006167	022054		ROL DSKNOR
1361	003640	052767	004000	022016	BIS #B11, FLAG ; SET MULTI DRIVE FLAG
1362	003646	005767	022042		TST DSKNOR ; WERE ANY UNITS AVAILABLE?
1363	003652	100011			BPL 1\$; BRANCH IF YES
1364	003654	004567	175404		JSR R5, PRINTS ; PRINT MESSAGE
1365	003660	024240			MES20
1366	003662	013701	000042		MOV #42, R1 ; ABORT - NO UNITS AVAILABLE
1367	003666	005067	174150		CLR 42 ; SET ABORT FLAG
1368	003672	000167	010650		JMP MEXIT
1369	003676	032767	004000	021760	1\$: BIT #B11, FLAG ; ARE WE IN MULTI DRIVE MODE?
1370	003704	001422			BEQ EXMFLG ; BRANCH IF NO
1371	003706	004567	175370		JSR R5, PRNTPS ; FORCE PRINT THE MESSAGE
1372	003712	024107			MES11
1373	003714	016767	021744	022004	MOV FLAG, ACNVX ; TELL OPERATOR THE UNIT UNDER TEST
1374	003722	006067	022000		ROR ACNVX
1375	003726	006067	021774		ROR ACNVX
1376	003732	042767	177770	021766	BIC #177770, ACNVX
1377	003740	016767	021762	175614	MOV ACNVX, TTY
1378	003746	004767	175426		JSR PC, PRINTB ; FORCE TYPE LOCATION - SUPPRESS ZEROS
1379	003752	032737	000010	177570	EXMFLG: BIT #B3, #SWR ; RUN SELECTED TEST?
1380	003760	001410			BEQ 1\$; BRANCH IF NO
1381	003762	013700	177570		MOV #SWR, RO ; GET SWITCH SETTINGS
1382	003766	042700	177770		BIC #177770, RO
1383	003772	000241			CLC
1384	003774	006100			ROL RO
1385	003776	000170	004016		JMP ATSTTBL(RO) ; GO TO SELECTED TEST
1386	004002	032767	002000	021654	1\$: BIT #B10, FLAG ; DATA TEST ONLY?
1387	004010	001412			BEQ ADT1 ; NO
1388	004012	000167	006312		JMP DATAT ; DO DATA TEST
1389					

003

1390 004016 004036
 1391 004020 004514
 1392 004022 005576
 1393 004024 006242
 1394 004026 010554
 1395 004030 012330
 1396 004032 013424
 1397 004034 014562

TSTTBL: ADT1
 ADT2
 ADT3
 WRCK
 MEMTST
 DATAT
 RANEX
 PFTST

.SBTTL ***** TEST 0 *****

; IN THIS TEST THE PROGRAM SEEKS FROM 0 TO N AND THEN BACK
 ; TO 0. N STARTS AT ZERO THEN INCREMENTS TO 1 AND UP THRU 625
 ; DONE IS TIMED OUT, SELECTED UNIT CYLINDER ADDRESS IS TESTED, SEEK UNDERWAY
 ; IS CHECK, AND THE ATTENTION FLAG IS TESTED.

1409 004036 005067 021666
 1410 004042 004567 175216
 1411 004046 023766
 1412 004050 016767 021654 175504
 1413 004056 004767 175274
 1414 004062 012777 004502 021544
 1415 004070 012777 000340 021540
 1416 004076 004567 012534
 1417 004102 005067 021570
 1418 004106 005067 021626
 1419 004112 005067 021624
 1420 004116 012737 000200 177776
 1421 004124 117777 021530 021526
 1422 004132 016777 021540 021510
 1423 004140 005067 000346
 1424 004144 052777 020011 021466
 1425 004152 012700 000025
 1426 004156 005300
 1427 004160 001376
 1428 004162 105777 021452
 1429 004166 100402
 1430 004170 104400
 1431 004172 000467
 1432 004174 005767 021540
 1433 004200 001406
 1434 004202 032777 002000 021450
 1435 004210 001002
 1436 004212 104400
 1437 004214 000456
 1438 004216 005000
 1439 004220 005200
 1440 004222 005777 021432
 1441 004226 100414
 1442 004230 005237 025746
 1443 004234 005237 025746
 1444 004240 005337 025746
 1445 004244 005337 025746

ADT1: CLR TESTNO
 JSR R5,PRINTS ;PRINT MESSAGE
 MES6
 MOV TESTNO,TTY
 JSR PC,PRINTS ;TYPE LOCATION-SUPPRESS ZEROS
 RADT1: MOV #INTCK,VECTOR ;SET UP DISK VECTOR
 MOV #340,STATUS
 JSR R5,DSKNOS ;SELECT UNIT
 CLR CYLINDER
 CLR WORK2 ;CYLINDER COUNTER
 CLR WORK3 ;POINTER
 MOV #PRI4,PSW ;ALLOW INTERRUPTS
 9\$: MOVB ARPDS,ARPDS ;CLEAR ATTENTION BITS
 MOV CYLINDER,ARPCA ;SET CYLINDER REGISTER
 CLR INTFLG ;CLEAR INTERRUPT FLAG
 BIS #20011,ARPCS ;SEEK AND ENABLE ATTN INTERRUPT
 MOV #25,R0
 1\$: DEC R0 ;DELAY FOR DONE TO SET
 BNE 1\$
 TSTB ARPCS ;TEST FOR DONE
 BMI 2\$;BRANCH DONE SET
 HLT ;DONE DID NOT SET AFTER SEEK
 BR 8\$
 2\$: TST WORK2 ;DON'T TEST SEEK UNDERWAY
 BEQ 3\$;IF FIRST TIME THRU
 BIT #B10,ARPDS ;DID SEEK UNDERWAY SET?
 BNE 3\$;BRANCH IF YES
 HLT ;SEEK UNDERWAY DID NOT SET
 BR 9\$
 3\$: CLR R0
 5\$: INC R0 ;TIMEOUT UNIT READY
 TST ARPDS ;IS UNIT READY?
 BMI 6\$;BRANCH IF YES
 INC #CYLA
 INC #CYLA
 DEC #CYLA
 DEC #CYLA

003

1446	004250	005700			TST	RO		; TIMEOUT?
1447	004252	001362			BNE	5\$		; BRANCH IF NO
1448	004254	104400			HLT			; READY DID NOT SET AFTER SEEK
1449	004256	000435			BR	8\$		
1450	004260	005767	000226		TST	INTFLG		; DID INTERRUPT OCCUR?
1451	004264	001002			BNE	12\$		; BRANCH IF YES
1452	004266	104400			HLT			; INTERRUPT DID NOT OCCUR ON ATTENTION BIT
1453	004270	000430			BR	8\$		
1454	004272	004767	012546		JSR	PC, GATTN		; DETERMINE ATTENTION BIT
1455	004276	036777	012572	021354	BIT	ATTN, ARPDS		; IS ATTENTION BIT SET?
1456	004304	001002			BNE	7\$		; BRANCH IF YES
1457	004306	104400			HLT			; ATTENTION BIT DID NOT SET
1458	004310	000420			BR	8\$		
1459	004312	026777	021360	021342	CMP	CYLINDER, ASUCA		; IS SUCA CORRECT?
1460	004320	001410			BEQ	11\$		
1461	004322	016767	021350	013474	MOV	CYLINDER, EXP\$		; EXPECTED RESULTS
1462	004330	017767	021326	013470	MOV	ASUCA, RECS		; RECEIVED RESULTS
1463	004336	104401			HLT	+1		; CONTETNS OF SUCA INCORRECT
1464	004340	000404			BR	8\$		
1465	004342	005777	021272		TST	ARPCS		; ANY DEVICE ERRORS
1466	004346	100001			BPL	8\$		; BRANCH IF NO
1467	004350	104400			HLT			; DEVICE ERROR AFTER SEEK OPERATION
1468	004352	032777	004000	021300	BIT	#B11, ARPDS		; SEEK INCOMPLETE ERROR?
1469	004360	001412			BEQ	4\$		; BRANCH IF NO
1470	004362	112777	000015	021250	MOVB	#15, ARPCS		; ISSUE HOME COMMAND
1471	004370	105777	021244		TSTB	ARPCS		; WAIT FOR DONE
1472	004374	100375			BPL	-4		
1473	004376	032777	100000	021254	BIT	#B15, ARPDS		; WAIT FOR UNIT READY
1474	004404	001774			BEQ	13\$		
1475	004406	012767	004124	174366	MOV	#9\$, LAD		; SET UP LOOP
1476	004414	104000			SCOPE			
1477	004416	005767	021320		TST	WORK3		; SEEK CYLINDER ZERO?
1478	004422	100411			BMI	10\$		; BRANCH IF YES
1479	004424	005267	021310		INC	WORK2		; UPDATE CYLINDER
1480	004430	016767	021304	021240	MOV	WORK2, CYLINDER		
1481	004436	052767	100000	021276	BIS	#B15, WORK3		; SET SEEK ZERO FLAG
1482	004444	000627			BR	9\$		
1483	004446	005067	021224		CLR	CYLINDER		
1484	004452	005067	021264		CLR	WORK3		; CLEAR SEEK ZERO FLAG
1485	004456	022767	000625	021254	CMP	#625, WORK2		; HAS LAST CYLINDER BEEN REACHED?
1486	004464	001217			BNE	9\$		; BRANCH IF NO
1487	004466	032737	002000	177570	BIT	#B10, ASWR		; REPEAT TEST
1488	004474	001407			BEQ	ADT2		; NO-GO TO NEXT
1489	004476	000167	177360		JMP	RADT1		; YES
1490								
1491								
1492	004502	012767	000001	000002	INTCK: MOV	#1, INTFLG		; SET INTERRUPT FLAG
1493	004510	000002			RTI			
1494	004512	000000			INTFLG: 0			
1495								
1496					.SBTTL ***** TEST 1 *****			
1497								
1498					; WRITE 5000 (OCTAL) WORDS IN TEN SECTORS ON EACH TRACK. THE FIRST			
1499					; WORD OF EACH SECTOR IS THE CYLINDER NUMBER AND THE REMAINING WORDS CONTAIN			
1500					; THE HEAD AND SECTOR ADDRESS. THEN EACH SECTOR IS READ BACK TEN AT A TIME AND			
1501					; COMPARED. IF THE FIRST WORD OF A SECTOR DOES NOT COMPARE, THE WRONG			

E03

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 29
 DZRPBE.P11 29-APR-77 14:41 ***** TEST 1 *****

```

1502                                     :CYLINDER WAS PROBABLY SELECTED.  A NON COMPARE ON THE FIRST
1503                                     :WORD IS INDICATED BY TYPING "CYL" AFTER THE ERRING DATA.
1504                                     :IF ANY OTHER WORD FAILS THE WRONG HEAD OR SECTOR WAS
1505                                     :SELECTED.  THE RIGHT HALF OF THE DATA TYPED EQUALS THE SECTOR
1506                                     :AND THE LEFT HALF INDICATES THE HEAD.
1507
1508 004514 012767 000001 021206 ADT2:  MOV    #1,TESTNO
1509 004522 004567 174536          JSR    R5,PRINT$      ;PRINT MESSAGE
1510 004526 023766          MES6
1511 004530 016767 021174 175024  MOV    TESTNO,TTY
1512 004536 004767 174614          JSR    PC,PRINT$      ;TYPE LOCATION-SUPPRESS ZEROS
1513 004542 005067 015202          CLR    MEX           ;CLEAR DRIVE EXTENDED MEMORY BITS
1514 004546 004567 012064          JSR    R5,DSKNOS      ;SELECT THE DRIVE
1515 004552 052777 000015 021060 1$:  BIS    #15,DRPCS      ;SEEK HOME
1516 004560 012700 000025          MOV    #25,R0
1517 004564 005300          2$:  DEC    R0              ;GIVE DONE A CHANCE TO SET
1518 004566 001376          BNE    2$
1519 004570 105777 021044          TSTB   DRPCS
1520 004574 100402          BMI    3$              ;IS DONE SET?
1521 004576 104400          HLT
1522 004600 000425          BR        6$              ;YES-BRANCH
1523 004602 005000          3$:  CLR    R0              ;DONE DID NOT SET AFTER A SEEK HOME
1524 004604 005200          5$:  INC    R0              ;CHECK FOR LOOPING
1525 004606 005777 021046          TST    DRPDS
1526 004612 100414          BMI    4$              ;IS UNIT READY?
1527 004614 005237 025746          INC    @CYLA
1528 004620 005237 025746          INC    @CYLA
1529 004624 005337 025746          DEC    @CYLA
1530 004630 005337 025746          DEC    @CYLA
1531 004634 005700          TST    R0
1532 004636 001362          BNE    5$              ;HAS UNIT TIMED OUT
1533 004640 104400          HLT
1534 004642 000404          BR        6$              ;NO-BRANCH
1535 004644 005777 020770          4$:  TST    DRPCS
1536 004650 100001          BPL    6$              ;READY DID NOT SET AFTER HOME SEEK
1537 004652 104400          HLT
1538 004654 012767 004552 174120 6$:  MOV    #1$,LAD      ;CHECK FOR LOOPING
1539 004662 104000          SCOPE
1540 004664 012767 005000 021002  MOV    #5000,WRDCT
1541 004672 012767 025762 021006  MOV    #OUTBUF,BUF
1542 004700 005067 020774          CLR    DMA
1543 004704 005067 020766          CLR    CYLINDER
1544 004710 012700 025762          SEABUF: MOV    #OUTBUF,R0      ;GET BUFFER STARTING ADDR
1545 004714 012701 000400          21$:  MOV    #400,R1      ;SECTOR COUNT
1546 004720 016720 020752          MOV    CYLINDER,(R0)+
1547 004724 005301          DEC    R1
1548 004726 016720 020746          1$:  MOV    DMA,(R0)+
1549 004732 005301          DEC    R1
1550 004734 001374          BNE    1$
1551 004736 122767 000011 020734  CMPB   #11,DMA
1552 004744 001403          BEQ    22$
1553 004746 005267 020726          INC    DMA
1554 004752 000760          BR        21$
1555 004754 105067 020720          22$:  CLRB   DMA
1556 004760 004567 010216          4$:  JSR    R5,FUNCT      ;UPDATE SECTOR COUNT
1557 004764 000003          .WORD  3              ;WRITE TEN SECTORS

```


1558	004766	005000				CLR	RO	
1559	004770	005200			3\$:	INC	RO	
1560	004772	105777	020642			TSTB	ARPCS	: IS DONE SET?
1561	004776	100404				BMI	2\$	: YES BRANCH
1562	005000	005700				TST	RO	: TEST FOR TIMEOUT
1563	005002	001372				BNE	3\$	: BRANCH IF NO
1564	005004	104400				HLT		: DONE DID NOT SET AFTER WRITE
1565	005006	000501				BR	5\$	
1566	005010	005777	020624		2\$:	TST	ARPCS	: ANY DEVICE ERRORS?
1567	005014	100002				BPL	6\$	: BRANCH IF NO
1568	005016	104400				HLT		: RP11C STATUS ERROR AFTER WRITE
1569	005020	000474				BR	5\$	
1570	005022	005067	020706		6\$:	CLR	WORK	: INCREMENT FLAG
1571	005026	017767	020620	012772		MOV	ARPCA, RECS	: GET DISK ADDR
1572	005034	042767	177760	012764		BIC	#177760, RECS	: SAVE SECTOR ADDR
1573	005042	005767	012760			TST	RECS	
1574	005046	001404				BEQ	9\$	: BRANCH IF SECTOR = ZERO
1575	005050	005067	012750			CLR	EXPS	
1576	005054	104401				HLT	+1	: SECTOR ADDR IN RPDA DID NOT UPDATE
1577	005056	000455				BR	5\$	: PROPERLY AFTER A TEN SECTOR WRITE
1578	005060	117767	020570	012740	9\$:	MOVB	ARPCA1, RECS	: GET THE HEAD ADDR
1579	005066	116767	020607	012730		MOVB	DMA+1, EXPS	: SECTOR ADDR OUTPUTTED
1580	005074	122767	000023	012722		CMPB	#23, EXPS	: DID WE OUTPUT HEAD 23?
1581	005102	001005				BNE	7\$	: BRANCH IF NO
1582	005104	005067	012714			CLR	EXPS	: RESET HEAD ADDR
1583	005110	010667	020620			MOV	SP, WORK	: SET INCREMENT FLAG
1584	005114	000402				BR	8\$	
1585	005116	005267	012702		7\$:	INC	EXPS	
1586	005122	126767	012676	012676	8\$:	CMPB	EXPS, RECS	: IS DISK HEAD ADDR CORRECT?
1587	005130	001402				BEQ	12\$	: BRANCH IF YES
1588	005132	104401				HLT	+1	: HEAD ADDR IN RPDA WAS INCORRECT
1589	005134	000426				BR	5\$	: AFTER TEN SECTOR WRITE
1590	005136	017767	020506	012662	12\$:	MOV	ARPCA, RECS	: GET DISK CYLINDER ADDR
1591	005144	016767	020526	012652		MOV	CYLINDER, EXPS	
1592	005152	005767	020556			TST	WORK	: IS INCREMENT FLAG SET?
1593	005156	001410				BEQ	13\$	: BRANCH IF NO
1594	005160	005267	012640			INC	EXPS	
1595	005164	022767	000626	012632		CMP	#626, EXPS	: WAS IT LAST CYLINDER?
1596	005172	001002				BNE	13\$	: BRANCH IF NO
1597	005174	005067	012624			CLR	EXPS	
1598	005200	026767	012620	012620	13\$:	CMP	EXPS, RECS	: IS DISK CYLINDER ADDR CORRECT?
1599	005206	001401				BEQ	5\$	: BRANCH IF YES
1600	005210	104401				HLT	+1	: CYLINDER ADDR IN RPDA IS NOT
1601								: CORRECT AFTER TEN SECTOR WRITE
1602	005212	032777	004000	020440	5\$:	BIT	#B11, ARPCS	: SEEK INCOMPLETE ERROR?
1603	005220	001412				BEQ	10\$	: BRANCH IF NO
1604	005222	112777	000015	020410		MOVB	#15, ARPCS	: ISSUE HOME COMMAND
1605	005230	105777	020404			TSTB	ARPCS	: WAIT FOR DONE
1606	005234	100375				BPL	-4	
1607	005236	032777	100000	020414	11\$:	BIT	#B15, ARPCS	: WAIT FOR UNIT READY
1608	005244	001774				BEQ	11\$	
1609	005246	012767	004760	173526	10\$:	MOV	#4\$, LAD	: SETUP LOOP ADDR
1610	005254	104000				SCOPE		
1611	005256	004767	010744			JSR	PC, DISBUF	: SETUP NEXT DISK ADDR
1612	005262	000612				BR	SEABUF	: WRITE NEXT SECTOR
1613	005264	012767	005000	020402		MOV	#5000, WRDCT	: RESTORE WORD COUNT

1614	005272	052777	000015	020340	BIS	#15,ARPCS	;SEEK HOME
1615	005300	105777	020334		TSTB	ARPCS	
1616	005304	100375			BPL	-4	;WAIT FOR DONE AFTER SEEK HOME
1617	005306	005777	020346		TST	ARPCS	
1618	005312	100375			BPL	-4	;WAIT FOR DRIVE READY AFTER SEEK HOME
1619	005314	012700	025762	RDSECT:	MOV	#OUTBUF,RO	
1620	005320	012701	005000		MOV	#5000,R1	
1621	005324	005020		23\$:	CLR	(R0)+	;CLEAR THE BUFFER
1622	005326	005301			DEC	R1	
1623	005330	001375			BNE	23\$	
1624	005332	004567	007644		JSR	RS,FUNCT	;READ TEN SECTORS
1625	005336	000005		.WORD	5		
1626	005340	105777	020274		TSTB	ARPCS	
1627	005344	100375			BPL	-4	;WAIT FOR DONE AFTER READ
1628	005346	005777	020266		TST	ARPCS	;ANY ERRORS?
1629	005352	100006			BPL	ADHGT	;BRANCH NO ERRORS
1630	005354	104400			HLT		;STATUS ERROR AFTER A READ
1631	005356	032777	040000	020254	BIT	#B14,ARPCS	;WAS IT A DATA ERROR?
1632	005364	001401			BEQ	ADHGT	;IF YES GO COMPARE DATA
1633	005366	000446			BR	ADTER1	
1634	005370	012700	025762	ADHGT:	MOV	#OUTBUF,RO	
1635	005374	012701	000400	ADHGT1:	MOV	#400,R1	
1636	005400	026710	020272		CMP	CYLINDER,(0)	;IS CYLINDER WORD CORRECT?
1637	005404	001017			BNE	ADERC	;BRANCH IF NO
1638	005406	005720			TST	(0)+	
1639	005410	005301			DEC	R1	
1640	005412	026710	020262	SANHT:	CMP	DMA,(0)	;IS HEAD-SECTOR WORD CORRECT?
1641	005416	001016			BNE	ADERR	;BRANCH IF NO
1642	005420	005720			TST	(0)+	
1643	005422	005301			DEC	R1	
1644	005424	001372			BNE	SANHT	
1645	005426	122767	000011	020244	CMPB	#11,DMA	
1646	005434	001423			BEQ	ADTER1	
1647	005436	005267	020236		INC	DMA	
1648	005442	000754			BR	ADHGT1	
1649	005444	016767	020226	012352	ADERC:	MOV	CYLINDER,EXPS
1650	005452	000403			BR	ADERC1	;CORRECT DATA/ADDRESS
1651	005454	016767	020220	012342	ADERR:	MOV	DMA,EXPS
1652	005462	011067	012340		ADERC1:	MOV	(0),RECS
1653	005466	104401			HLT	+1	;CORRECT DATA/ADDRESS
1654	005470	022701	000400		CMP	#400,R1	;INCORRECT DATA
1655	005474	001003			BNE	ADTER1	;DATA COMPARE ERROR
1656	005476	004567	173562		JSR	RS,PRINTS	;WAS FIRST WORD INCORRECT?
1657	005502	024123			MES12		;BRANCH IF NO
1658	005504	105067	020170		CLRB	DMA	;PRINT MESSAGE
1659	005510	032777	004000	020142	ADTER1:	CLRB	;WRONG CYLINDER PROBABLY SELECTED
1660	005516	001412			BIT	#B11,ARPCS	;SEEK INCOMPLETE ERROR?
1661	005520	112777	000015	020112	BEQ	1\$	;BRANCH IF NO
1662	005526	105777	020106		MOV	#15,ARPCS	;ISSUE HOME COMMAND
1663	005532	100375			TSTB	ARPCS	;WAIT FOR DONE
1664	005534	032777	100000	020116	BPL	-4	
1665	005542	001774			BIT	#B15,ARPCS	;WAIT FOR UNIT READY
1666	005544	012767	005314	173230	BEQ	2\$	
1667	005552	104000			MOV	#RDSECT,LAD	;SETUP LOOP
1668	005554	004767	010446		SCOPE		
1669	005560	000655			JSR	PC,DISBUF	;SETUP NEXT DISK ADDRESS
					BR	RDSECT	;CHECK NEXT SECTOR

H03

 RPIIC RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 32
 DZAPBE.PII 29-APR-77 14:41 ***** TEST 1 *****

```

1670 005562 032737 002000 177570      BIT      #B10,2#SWR      ;LOOP ON TEST?
1671 005570 001402                BEQ      ADT3          ;BRANCH IF NO
1672 005572 000167 176750                JMP      RADT2
1673                                .SBTTL ***** TEST 2 *****
1674
1675                                ;WRITE THE FIRST WORD OF EACH CYLINDER WITH THE CYLINDER ADDRESS.
1676                                ;THEN SEEK FROM 625 TO 0. THEN SEEK TO 624 AND BACK TO 1
1677                                ;UNTIL THE NUMBERS CROSS. AFTER EACH SEEK VERIFY THE POSITION
1678                                ;BY READING THE FIRST WORD ON THE CYLINDER AND COMPARING
1679                                ;AGAINST CYLINDER REQUESTED.
1680
1681 005576 012767 000002 020124 ADT3:  MOV      #2,TESTNO
1682 005604 004567 173454                JSR      RS,PRINT$      ;PRINT MESSAGE
1683 005610 023766                MES6
1684 005612 016767 020112 173742                MOV      TESTNO,TTY
1685 005620 004767 173532                JSR      PC,PRINTS      ;TYPE LOCATION-SUPPRESS ZEROS
1686 005624 005067 014120                CLR      MEX          ;CLEAR DRIVE EXTENDED MEMORY BITS
1687 005630 004767 011054                RADT3: JSR      PC,INIT      ;INITIALIZE VECTORS
1688 005634 004567 010776                JSR      RS,DSKNOS      ;SELECT UNIT
1689 005640 005067 020034                CLR      DMA
1690 005644 005067 020026                CLR      CYLINDER
1691 005650 012767 000001 020016                MOV      #1,WRDCT      ;SET UP WORD COUNT
1692 005656 012767 025762 020022                MOV      #OUTBUF,BUF   ;SETUP BUFFER ADDR
1693 005664 016737 020006 025762 WRCYL: MOV      CYLINDER,2#OUTBUF ;INSERT PATTERN
1694 005672 004567 007304                JSR      RS,FUNCT      ;WRITE PATTERN ON FIRST SECTOR
1695                                .WORD 3          ;OF CYLINDER
1696 005700 105777 017734                TSTB     2RPCS          ;WAIT FOR DONE
1697 005704 100375                BPL      -4
1698 005706 005777 017726                TST      2RPCS          ;AND ERRORS?
1699 005712 100002                BPL      1$            ;BRANCH IF NO
1700 005714 104400                HLT
1701 005716 000407                BR      2$
1702 005720 022767 000625 017750 1$:  CMP      #625,CYLINDER ;ALL CYLINDERS WRITTEN?
1703 005726 001403                BEQ      2$            ;BRANCH IF YES
1704 005730 005267 017742                INC      CYLINDER
1705 005734 000753                BR      WRCYL
1706 005736 032777 004000 017714 2$:  BIT      #B11,2RPCS      ;SEEK INCOMPLETE ERROR?
1707 005744 001412                BEQ      3$            ;BRANCH IF NO
1708 005746 112777 000015 017664                MOVB     #15,2RPCS      ;ISSUE HOME COMMAND
1709 005754 105777 017660                TSTB     2RPCS          ;WAIT FOR DONE
1710 005760 100375                BPL      -4
1711 005762 032777 100000 017670 4$:  BIT      #B15,2RPCS      ;WAIT FOR UNIT READY
1712 005770 001774                BEQ      4$
1713 005772 012767 005664 173002 3$:  MOV      #WRCYL,LAD     ;SETUP UP LOOP
1714 006000 104000                SCOPE
1715 006002 005067 017670                CLR      CYLINDER
1716 006006 012767 000624 017724                MOV      #624,WORK2
1717 006014 005067 017722                CLR      WORK3
1718 006020 005067 017720                CLR      WORK4          ;INC - DEC FLAG
1719 006024 016767 017672 017654 ADT32: MOV      INBUF,BUF
1720 006032 004567 007144                JSR      RS,FUNCT      ;READ THE FIRST WORD OF THE
1721 006036 000005                .WORD 5          ;CYLINDER
1722 006040 105777 017574                TSTB     2RPCS          ;WAIT FOR DONE AFTER READ
1723 006044 100375                BPL      -4
1724 006046 005777 017566                TST      2RPCS          ;ANY ERRORS?
1725 006052 100002                BPL      3$            ;BRANCH IF NO

```


RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 33
 DZRPBE.P11 29-APR-77 14:41 ***** TEST 2 *****

1726	006054	104400				HLT			;ERROR AFTER READING ONE WORD
1727	006056	000413				BR	5\$		
1728	006060	027767	017636	017610	3\$:	CMP	2INBUF,CYLINDER		;COMPARE DATA READ AGAINST CYLINDER
1729	006066	001407				BEQ	5\$		;BRANCH IF EQUAL
1730	006070	016767	017602	011726		MOV	CYLINDER,EXP\$		;CORRECT DATA
1731	006076	017767	017620	011722		MOV	2INBUF,REC\$		;INCORRECT DATA
1732	006104	104401				HLT	+1		;DATA COMPARE ERROR-PROBABLY WENT
1733									;TO THE WRONG CYLINDER
1734	006106	032777	004000	017544	5\$:	BIT	#B11,2RPDS		;SEEK INCOMPLETE ERROR?
1735	006114	001412				BEQ	2\$		;BRANCH IF NO
1736	006116	112777	000015	017514		MOV	#15,2RPCS		;ISSUE A HOME COMMAND
1737	006124	105777	017510			TST	2RPCS		;WAIT FOR DONE
1738	006130	100375				BPL	.-4		
1739	006132	032777	100000	017520	4\$:	BIT	#B15,2RPDS		;WAIT FOR UNIT READY
1740	006140	001774				BEQ	4\$		
1741	006142	012767	006024	172632	2\$:	MOV	#ADT32,LAD		;SETUP LOOP
1742	006150	104000				SCOPE			
1743	006152	005767	017566			TST	WORK4		;INC - DEC FLAG
1744	006156	100411				BMI	1\$		
1745	006160	005267	017556			INC	WORK3		;UPDATE LOW COUNT
1746	006164	016767	017550	017504		MOV	WORK2,CYLINDER		
1747	006172	052767	100000	017544		BIS	#B15,WORK4		;SET DECREMENT FLAG
1748	006200	000711				BR	ADT32		
1749	006202	005367	017532		1\$:	DEC	WORK2		;DECREMENT HIGH COUNT
1750	006206	005067	017532			CLR	WORK4		;CLEAR FLAG
1751	006212	016767	017524	017456		MOV	WORK3,CYLINDER		
1752	006220	005767	017514			TST	WORK2		;DONE YET
1753	006224	001277				BNE	ADT32		;BRANCH-NO
1754	006226	032737	002000	177570		BIT	#B10,2\$SWR		;LOOP ON TEST?
1755	006234	001402				BEQ	WRCK		;NO
1756	006236	000167	177366			JMP	RADT3		;YES
1757									
1758									
1759									
1760									
1761									
1762									
1763									
1764									
1765									
1766									
1767									
1768	006242	012767	000003	017460		WRCK:	MOV	#3,TESTNO	
1769	006250	004567	173010				JSR	R5,PRINT\$	
1770	006254	023766					MES6		;PRINT MESSAGE
1771	006256	016767	017446	173276			MOV	TESTNO,TTY	
1772	006264	004767	173066				JSR	PC,PRINTS	
1773	006270	005067	013454				CLR	MEX	;TYPE LOCATION-SUPPRESS ZEROS
1774	006274	004767	010410			RWRCK:	JSR	PC,INIT	;CLEAR EXTENDED MEMORY BITS IN CONTROLER
1775	006300	004567	010332				JSR	R5,DSKNOS	;INITIALIZE
1776	006304	005067	017366				CLR	CYLINDER	;SELECT UNIT
1777	006310	005067	017364				CLR	DMA	
1778	006314	005000					CLR	RO	;PATTERN FLAG
1779	006316	012701	000001				MOV	#1,R1	;STARTING PATTERN
1780	006322	012767	025762	017356			MOV	#0UTBUF,BUF	;SETUP OUTPUT BUFFER
1781	006330	012767	000400	017336			MOV	#400,WRDCT	;SETUP WORDCOUNT

.SBTTL ***** TEST 3 *****

;THIS ROUTINE VERIFIES THE WRITE CHECK LOGIC AND
 ;THE ABILITY OF THE HARDWARE TO FILL THE REMAINDER
 ;OF A SECTOR WITH ZEROS WHEN A PARTIAL SECTOR
 ;IS WRITTEN. IN THE WRITE CHECK PORTION A
 ;FLOATING ONE AND FLOATING ZERO PATTERNS ARE USED
 ;TO TEST THE COMPARE LOGIC AND TO VERIFY THAT A
 ;WRITE CHECK ERROR WILL OCCUR.

1782	006336	005002		23\$:	CLR	R2	
1783	006340	010162	025762	1\$:	MOV	R1,OUTBUF(R2)	;GENERATE TEST PATTERN
1784	006344	005722			TST	(R2)+	;UPDATE MODIFIER
1785	006346	022702	000400		CMP	#400,R2	;HAS BUFFER BEEN FILLED?
1786	006352	001372			BNE	1\$	;BRANCH IF NO
1787	006354	004567	006622		JSR	R5,FUNCT	;WRITE PATTERN
1788	006360	000003		.WORD	3		
1789	006362	105777	017252		TSTB	3RPCS	;WAIT FOR DONE
1790	006366	100375			BPL	-4	
1791	006370	005777	017244		TST	3RPCS	;ANY DEVICE ERRORS?
1792	006374	100002			BPL	2\$	;BRANCH IF NO
1793	006376	104400			HLT		;ERROR AFTER WRITING ONE SECTOR
1794	006400	000475			BR	3\$	
1795	006402	004567	006574	2\$:	JSR	R5,FUNCT	;WRITE CHECK THE DATA
1796	006406	000007		.WORD	7		
1797	006410	105777	017224		TSTB	3RPCS	;WAIT FOR DONE
1798	006414	100375			BPL	-4	
1799	006416	005777	017216		TST	3RPCS	;ANY DEVICE ERRORS?
1800	006422	100012			BPL	4\$	;BRANCH IF NO
1801	006424	104400			HLT		;ERROR AFTER WRITE CHECK OPERATION
1802	006426	004567	172632		JSR	R5,PRINT\$	;PRINT MESSAGE
1803	006432	024070			MES10		
1804	006434	016767	017322	173120	MOV	OUTBUF,TTY	
1805	006442	004767	172676		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
1806	006446	000452			BR	3\$	
1807	006450	005700		4\$:	TST	R0	;ARE WE FLOATING A ONE?
1808	006452	001411			BEQ	20\$	;BRANCH IF YES
1809	006454	005002			CLR	R2	;FILL BUFFER WITH ONES
1810	006456	012762	177777	025762	21\$:	MOV	#177777,OUTBUF(R2)
1811	006464	005722			TST	(R2)+	
1812	006466	022702	000400		CMP	#400,R2	
1813	006472	001371			BNE	21\$	
1814	006474	000407			BR	22\$	
1815	006476	005002		20\$:	CLR	R2	
1816	006500	005062	025762	5\$:	CLR	OUTBUF(R2)	;CLEAR OUTPUT BUFFER
1817	006504	005722			TST	(R2)+	
1818	006506	022702	000400		CMP	#400,R2	;ENTIRE BUFFER CLEAR?
1819	006512	001372			BNE	5\$	;BRANCH IF NO
1820	006514	004567	006462	22\$:	JSR	R5,FUNCT	;WRITE CHECK THE DATA AND
1821	006520	000007		.WORD	7		;EXPECT AN ERROR
1822	006522	105777	017112		TSTB	3RPCS	;WAIT FOR DONE
1823	006526	100375			BPL	-4	
1824	006530	032777	000010	017120	BIT	#B3,3RPER	;IS WRITE CHECK ERROR SET?
1825	006536	001011			BNE	6\$	;BRANCH IF YES
1826	006540	104400			HLT		;WRITE CHECK ERROR DID NOT SET
1827	006542	004567	172516		JSR	R5,PRINT\$	;PRINT MESSAGE
1828	006546	024070			MES10		
1829	006550	010167	173006		MOV	R1,TTY	
1830	006554	004767	172564		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
1831	006560	000405			BR	3\$	
1832	006562	005777	017052	6\$:	TST	3RPCS	;DID ERROR FLAG SET?
1833	006566	100402			BMI	3\$	;BRANCH IF YES
1834	006570	104400			HLT		;ERROR FLAG DID NOT SET AFTER WRITE CHECK ERROR
1835	006572	000400			BR	3\$	
1836	006574	012767	006336	172200	3\$:	MOV	#23\$,LAD
1837	006602	104000			SCOPE		;SETUP LOOP ADDR

K03

1838	006604	005700		TST	RO		;ARE WE FLOATING A ONE?
1839	006606	001013		BNE	9\$		;BRANCH IF NO
1840	006610	000241		CLC			
1841	006612	006101		ROL	R1		;ROTATE PATTERN
1842	006614	103402		BCS	10\$		;BRANCH IF COMPLETE
1843	006616	000167	177514	JMP	23\$		
1844	006622	012700	000001	10\$: MOV	#1,RO		;SET PATTERN FLAG
1845	006626	012701	077777	MOV	#077777,R1		;SET NEW PATTERN IN R1
1846	006632	000167	177500	JMP	23\$		
1847	006636	000241		9\$: CLC			
1848	006640	006201		ASR	R1		;ROTATE FLOATING ZERO PATTERN
1849	006642	052701	100000	BIS	#B15,R1		
1850	006646	103002		BCC	PATFIL		;HAS ZERO BEEN FLOATED
1851	006650	000167	177462	JMP	23\$		;JUMP IF NO
1852							
1853							
1854							
1855							
1856							
1857							
1858							
1859							
1860							
1861	006654	012701	025762	PATFIL: MOV	#OUTBUF,R1		
1862	006660	012700	177777	MOV	#177777,RO		
1863	006664	012702	000400	MOV	#400,R2		
1864	006670	010021		1\$: MOV	RO,(R1)+		;GENERATE ALL ONES PATTERN
1865	006672	005302		DEC	R2		
1866	006674	001375		BNE	1\$		
1867	006676	004567	006300	JSR	R5,FUNCT		;WRITE SECTOR WITH ONES
1868	006702	000003		.WORD	3		
1869	006704	105777	016730	TSTB	DRPCS		;WAIT FOR DONE
1870	006710	100375		BPL	-4		
1871	006712	005777	016722	TST	DRPCS		;ANY DEVICE ERRORS
1872	006716	100002		BPL	2\$		;BRANCH IF NO
1873	006720	104400		HLT			;ERROR AFTER WRITING ONE SECTOR ALL 1'S
1874	006722	000473		BR	3\$		
1875	006724	012767	000002	2\$: MOV	#2,WRDCT		;SETUP FOR TWO WORD WRITE
1876	006732	004567	006244	JSR	R5,FUNCT		;WRITE TWO WORD
1877	006736	000003		.WORD	3		
1878	006740	105777	016674	TSTB	DRPCS		;WAIT FOR DONE
1879	006744	100375		BPL	-4		
1880	006746	005777	016666	TST	DRPCS		;ANY ERRORS?
1881	006752	100002		BPL	4\$		;BRANCH IF NO
1882	006754	104400		HLT			;ERROR ON ONE WORD WRITE
1883	006756	000455		BR	3\$		
1884	006760	012767	000400	4\$: MOV	#400,WRDCT		;SETUP WORD COUNT
1885	006766	004567	006210	JSR	R5,FUNCT		;READ SECTOR
1886	006772	000005		.WORD	5		
1887	006774	105777	016640	TSTB	DRPCS		;WAIT FOR DONE
1888	007000	100375		BPL	-4		
1889	007002	005777	016632	TST	DRPCS		;ANY ERRORS
1890	007006	100006		BPL	5\$		;BRANCH IF NO
1891	007010	104400		HLT			;ERROR AFTER READING ONE SECTOR
1892	007012	032777	040000	BIT	#B14,DRPCS		;WAS IT A DATA ERROR?
1893	007020	001401		BEQ	5\$		;BRANCH IF YES


```

1894 007022 000433 BR 3$
1895 007024 022767 177777 016730 5$: CMP #177777,OUTBUF ;COMPARE FIRST WORD SHOULD BE ONES
1896 007032 001410 BEQ 6$ ;BRANCH IF OK
1897 007034 012767 177777 010762 MOV #177777,EXP$
1898 007042 016767 016714 010756 MOV OUTBUF,REC$
1899 007050 104401 HLT +1 ;DATA COMPARE ERROR ON FIRST
1900 007052 000417 BR 3$ ;WORD READ
1901 007054 012700 025766 6$: MOV #OUTBUF+4,R0
1902 007060 012701 000374 MOV #374,R1
1903 007064 005720 8$: TST (R0)+ ;REMAINDER OF SECTOR SHOULD BE CLEAR
1904 007066 001003 BNE 7$ ;BRANCH IF NO
1905 007070 005301 DEC R1
1906 007072 001374 BNE 8$
1907 007074 000406 BR 3$
1908 007076 016067 177776 010722 7$: MOV -2(R0),REC$
1909 007104 005067 010714 CLR EXP$
1910 007110 104401 HLT +1 ;DATA FOUND IN AREA OF SECTOR
1911 ;WHICH SHOULD HAVE BEEN CLEARED
1912 ;BY A ONE WORD WRITE
1913 007112 012767 006654 171662 3$: MOV #PATFIL,LAD ;SET UP LOOP ADDR
1914 007120 104000 SCOPE
1915
1916 ;CHECK THE SETTING OF EOP WHEN TRYING TO WRITE BEYOND
1917 ;THE LIMITS OF THE PACK. THE FIRST SECTOR OF THE PACK IS
1918 ;WRITTEN WITH ZEROS. THEN A TWO SECTOR WRITE OF ALL
1919 ;ONE'S IS ISSUED FOR CYLINDER 625, HEAD 23, AND SECTOR 11.
1920 ;EOP AND ERROR BITS SHOULD SET. THE FIRST SECTOR OF THE
1921 ;PACK IS CHECKED TO MAKE SURE IT IS STILL ZERO.
1922
1923
1924 007122 005067 016552 EOPTST: CLR DMA ;CLEAR DISK ADDRESS
1925 007126 005067 016544 CLR CYLINDER
1926 007132 012767 000400 016534 MOV #400,WRDCT ;SET WORDCOUNT TO ONE SECTOR
1927 007140 012767 025762 016540 MOV #OUTBUF,BUF ;SETUP OUTPUT BUFFER
1928 007146 005001 CLR R1
1929 007150 005061 025762 1$: CLR OUTBUF(R1) ;CLEAR THE OUTPUT BUFFER
1930 007154 005721 TST (R1)+
1931 007156 022701 000400 CMP #400,R1
1932 007162 001372 BNE 1$
1933 007164 004567 006012 JSR R5,FUNCT ;WRITE SECTOR ZERO WITH ZEROS
1934 007170 000003 .WORD 3
1935 007172 105777 016442 TSTB #RPCS ;WAIT FOR DONE
1936 007176 100375 BPL -4
1937 007200 005777 016434 TST #RPCS ;ANY DEVICE ERRORS
1938 007204 100002 BPL 2$ ;BRANCH IF NO
1939 007206 104400 HLT ;ERROR AFTER WRITING SECTOR ZERO WITH ZEROS
1940 007210 000502 BR 3$
1941 007212 012767 001000 016454 2$: MOV #1000,WRDCT ;SET WORDCOUNT EQUAL TO TWO SECTORS
1942 007220 012767 000625 016450 MOV #625,CYLINDER ;SELECT CYLINDER 625
1943 007226 012767 000011 016444 MOV #11,DMA ;SELECT RECTOR 11
1944 007234 112767 000023 016437 MOVB #23,DMA+1 ;SELECT HEAD 23
1945 007242 012702 177777 MOV #177777,R2
1946 007246 005001 CLR R1
1947 007250 010261 025762 4$: MOV R2,OUTBUF(R1) ;SET OUTPUT BUFFER TO ONES
1948 007254 005721 TST (R1)+
1949 007256 022701 001000 CMP #1000,R1

```


1950	007262	001372			BNE	4\$		
1951	007264	004567	005712		JSR	RS,FUNCT		
1952	007270	000003		.WORD	3			:ISSUE TWO SECTOR WRITE TO
1953	007272	105777	016342		TSTB	ARPCS		:CYLINDER 625, HEAD 23, AND SECTOR 11
1954	007276	100375			BPL	-4		:WAIT FOR DONE
1955	007300	032777	000002	016350	BIT	#B1,ARPER		:DID EOP ERROR FLAG SET?
1956	007306	001002			BNE	5\$		:BRANCH IF SET
1957	007310	104400			HLT			:EOP ERROR FLAG DID NOT SET WHEN
1958	007312	000441			BR	3\$		:WRITE OPERATOR EXCEEDS THE PACK
1959	007314	032777	100000	016316	BIT	#B15,ARPCS		:DID THE ERROR FLAG SET?
1960	007322	001002			BNE	6\$		:BRANCH IF SET
1961	007324	104400			HLT			:ERROR DID NOT SET AFTER GENERATING
1962	007326	000433			BR	3\$		:EOP
1963	007330	012767	000002	016336	MOV	#2,WRDCT		:CLEAR THE DISK ADDRESS
1964	007336	005067	016336		CLR	DMA		
1965	007342	005067	016330		CLR	CYLINDER		
1966	007346	004567	005630		JSR	RS,FUNCT		:READ THE FIRST SECTOR OF THE PACK
1967	007352	000005		.WORD	5			:AND EXPECT TO FIND ZEROS
1968	007354	105777	016260		TSTB	ARPCS		:WAIT FOR READY
1969	007360	100375			BPL	-4		
1970	007362	005777	016252		TST	ARPCS		:WERE THERE ANY ERRORS?
1971	007366	100002			BPL	7\$		:BRANCH IF NO
1972	007370	104400			HLT			:ERROR ENCOUNTERED ON 2 WORD READ
1973	007372	000411			BR	3\$		:OF FIRST SECTOR ON THE PACK
1974	007374	016767	016362	010424	MOV	OUTBUF,RECS		:GET FIRST WORD OF BUFFER
1975	007402	005767	010420		TST	RECS		:DOES 1ST SECTOR STILL CONTAIN ZEROS?
1976	007406	001403			BEQ	3\$		:BRANCH IF YES
1977	007410	005067	010410		CLR	EXPS		
1978	007414	104401			HLT	+1		:CONTENTS OF THE FIRST SECTOR OF THE
1979								:PACK CHANGED AFTER FORCING EOP
1980								:ERROR, OPERATION PROBABLY
1981								:WRAPPED AROUND.
1982	007416	012767	007122	171356	MOV	#EOPST,LAD		
1983	007424	104000			SCOPE			
1984								
1985								
1986								:CHECK THE ABILITY OF THE RP11C TO GENERATE A
1987								:PROGRAM ERROR IF A COMMAND IS ISSUED WHILE THE
1988								:CONTROLLER IS BUSY.
1989								
1990	007426	012777	000001	016204	LOKOUT: MOV	#1,ARPCS		:CLEAR THE CONTROLLER
1991	007434	005067	016236		CLR	CYLINDER		:CLEAR THE DISK ADDRESS
1992	007440	005067	016234		CLR	DMA		
1993	007444	012767	002000	016222	MOV	#2000,WRDCT		:SETUP WORDCOUNT
1994	007452	004567	005524		JSR	RS,FUNCT		:ISSUE A READ COMMAND TO GET THE
1995	007456	000005		.WORD	5			:CONTROLLER BUSY
1996	007460	012700	000100		MOV	#100,RO		
1997	007464	005300		1\$:	DEC	RO		:ALLOW THE OPERATION TO PROCEED AWHILE
1998	007466	001376			BNE	1\$		
1999	007470	112777	000005	016142	MOVB	#5,ARPCS		:ISSUE READ COMMAND WHILE BUSY
2000	007476	105777	016136		TSTB	ARPCS		:WAIT FOR DONE
2001	007502	100375			BPL	-4		
2002	007504	032777	002000	016144	BIT	#B10,ARPER		:DID PROGRAM ERROR SET?
2003	007512	001002			BNE	2\$		:BRANCH IF SET
2004	007514	104400			HLT			:PROGRAM ERROR DID NOT SET WHEN A
2005								:READ COMMAND WAS ISSUED WHILE

NO3

```

2006                                     ;THE DEVICE WAS BUSY
2007 007516 000405                                     3$
2008 007520 032777 100000 016112 2$: BR 3$
2009 007526 001001                                     #B15,ARPCS ;DID THE ERROR FLAG SET?
2010 007530 104400                                     BNE 3$ ;BRANCH IF SET
2011                                     HLT ;ERROR FLAG DID NOT SET AFTER
2012 007532 012767 007426 171242 3$: MOV #LOKOUT,LAD ;PROGRAM ERROR
2013 007540 104000 SCOPE
2014
2015
2016                                     ;UNFORMAT THE FIRST SECTOR ON THE PACK. THEN READ IT BACK
2017                                     ;AND VERIFY THAT READ AND WRITE HEADER OPERATIONS WILL
2018                                     ;TRANSFER DATA CORRECTLY. NOW THAT THE HEADER IS MIS FORMATTED,
2019                                     ;ISSUE A WRITE COMMAND TO SECTOR ZERO. THIS SHOULD RESULT
2020                                     ;IN SETTING HEADER NOT FOUND. THEN REFORMAT SECTOR ZERO.
2021
2022 007542 012777 000001 016070 HNFCK: MOV #1,ARPCS ;CLEAR THE CONTROLLER
2023 007550 005067 016122 CLR CYLINDER ;CLEAR DISK ADDR
2024 007554 005067 016120 CLR DMA
2025 007560 012767 025762 016120 MOV #OUTBUF,BUF ;SETUP BUFFER ADDR
2026 007566 012700 025762 MOV #OUTBUF,RO
2027 007572 012720 000001 MOV #1,(RO)+
2028 007576 012720 000001 MOV #1,(RO)+
2029 007602 012720 000001 MOV #1,(RO)+
2030 007606 012767 000003 016060 MOV #3,WRDCT ;LOAD WORDCOUNT
2031 007614 004567 005362 JSR R5,FUNCT ;ISSUE WRITE HEADER COMMAND
2032 007620 014003 .WORD 14003 ;TO MISFORMAT SECTOR ZERO
2033 007622 105777 016012 TSTB ARPCS ;WAIT FOR READY
2034 007626 100375 BPL -4
2035 007630 005777 016004 TST ARPCS ;ANY ERRORS?
2036 007634 100002 BPL 1$ ;BRANCH IF NO
2037 007636 104400 HLT ;ERROR AFTER ISSUING A WRITE
2038                                     ;FORMAT COMMAND TO SECTOR ZERO
2039 007640 000440 BR 2$
2040 007642 004567 005334 1$: JSR R5,FUNCT ;ISSUE READ HEADER COMMAND
2041 007646 014005 .WORD 14005 ;TO SECTOR ZERO
2042 007650 105777 015764 TSTB ARPCS ;WAIT FOR READY
2043 007654 100375 BPL -4
2044 007656 005777 015756 TST ARPCS ;ANY ERRORS?
2045 007662 100005 BPL 3$ ;BRANCH IF NO
2046 007664 104400 HLT ;ERROR WHILE READING THE HEADER
2047                                     ;ON SECTOR ZERO
2048 007666 032777 040000 015744 BIT #B14,ARPCS ;HARD ERROR?
2049 007674 001022 BNE 2$ ;BRANCH IF YES
2050 007676 012767 000001 010120 3$: MOV #1,EXP$ ;EXPECTED RESULT
2051 007704 005000 CLR RO
2052 007706 026760 010112 025762 5$: CMP EXP$,OUTBUF(RO) ;CHECK DATA READ BACK
2053 007714 001006 BNE 4$ ;BRANCH ON NON COMPARE
2054 007716 062700 000002 ADD #2,RO ;UPDATE MODIFIER
2055 007722 022700 000006 CMP #6,RO ;END OF BUFFER?
2056 007726 001367 BNE 5$
2057 007730 000404 BR 2$
2058 007732 016067 025762 010066 4$: MOV OUTBUF(RO),RECS ;GET BAD DATA
2059 007740 104403 HLT +3 ;DATA DID NOT VERIFY AFTER READING
2060                                     ;THE HEADER OF SECTOR ZERO
2061 007742 012767 007542 171032 2$: MOV #HNFCK,LAD

```


2062	007750	104000				SCOPE		
2063	007752	004567	005224			HNF1: JSR	RS,FUNCT	:ISSUE WRITE TO SACTOR ZERO
2064	007756	000003				.WORD 3		
2065	007760	005000				CLR RO		
2066	007762	005200				INC RO		:TIMEOUT SETTING OF DONE
2067	007764	105777	015650			TSTB JRPCS		:DID DONE SET?
2068	007770	100414				BMI 15		
2069	007772	005237	025746			INC JRCYLA		
2070	007776	005237	025746			INC JRCYLA		
2071	010002	005337	025746			DEC JRCYLA		
2072	010006	005337	025746			DEC JRCYLA		
2073	010012	005700				TST RO		:TIMEOUT?
2074	010014	001362				BNE 25		:BRANCH IF NO
2075	010016	104400				HLT		:TIMED OUT WHILE WAITING FOR DONE
2076								:AFTER TRYING TO FORCE HEADER NOT FOUND
2077	010020	000427				BR 35		
2078	010022	032777	010000	015630	15:	BIT B12,JRPDS		:DID HEADER NOT FOUND SET?
2079	010030	001002				BNE 65		:BRANCH IF YES
2080	010032	104400				HLT		:HEADER NOT FOUND DID NOT SET
2081	010034	000421				BR 35		
2082	010036	032777	040000	015574	65:	BIT B14,JRPCS		:DID HARD ERROR SET?
2083	010044	001002				BNE 45		:BRANCH IF YES
2084	010046	104400				HLT		:HARD ERROR NOT SET AFTER HNF
2085	010050	000413				BR 35		
2086	010052	032777	100000	015560	45:	BIT B15,JRPCS		:DID THE ERROR BIT SET?
2087	010060	001002				BNE 55		:BRANCH IF YES
2088	010062	104400				HLT		:ERROR DID NOT SET AFTER HNF
2089	010064	000405				BR 35		
2090	010066	032777	000001	015562	55:	BIT B0,JRPER		:DID DISK ERROR SET?
2091	010074	001001				BNE 35		:BRANCH IF YES
2092	010076	104400				HLT		:DISK ERROR DID NOT SET AFTER HNF
2093	010100	012767	007752	170674	35:	MOV HNF1,LAD		
2094	010106	104000				SCOPE		
2095	010110	005000				HNF2: CLR RO		
2096	010112	005060	025762		15:	CLR OUTBUF(RO)		:CLEAR BUFFER
2097	010116	062700	000002			ADD #2,RO		:UPDATE MODIFIER
2098	010122	022700	000006			CMP #6,RO		:ENTIRE BUFFER CLEARED?
2099	010126	001371				BNE 15		:BRANCH IF NO
2100	010130	012767	000003	015536		MOV #3,WRDCT		:LOAD WORD COUNT
2101	010136	004567	005040			JSR RS,FUNCT		:ISSUE WRITE HEADER COMMAND
2102	010142	014003				.WORD 14003		
2103	010144	105777	015470			TSTB JRPCS		:WAIT FOR READY
2104	010150	100375				BPL -4		
2105	010152	005777	015462			TST JRPCS		:ANY ERRORS?
2106	010156	100001				BPL 25		
2107	010160	104400				HLT		:ERROR WHILE TRYING TO FORMAT SECTOR ZERO
2108	010162	012767	010110	170612	25:	MOV HNF2,LAD		
2109	010170	104000				SCOPE		
2110								
2111								:ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A WRITE
2112								:COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11C SHOULD
2113								:HOLD THE WRITE COMMAND TILL THE SEEK IS COMPLETE.
2114								
2115	010172	012777	000001	015440	SKTST:	MOV #1,JRPCS		:CLEAR THE CONTROLLER
2116	010200	004567	006432			JSR RS,DSKNOS		:SELECT THE UNIT
2117	010204	005077	015440			CLR JRPCA		:CLEAR THE CYLINDER ADDR REGISTER

2118	010210	005077	015436		CLR	DRPDA	;CLEAR THE DISK ADDR REGISTER
2119	010214	112777	000011	015416	MOVB	#11,DRPCS	;ISSUE A SEEK TO ZERO
2120	010222	105777	015412		TSTB	DRPCS	;WAIT FOR DONE
2121	010226	100375			BPL	.-4	
2122	010230	032777	100000	015422	1\$:	#B15,DRPDS	;WAIT FOR UNIT READY
2123	010236	001774			BEQ	1\$	
2124	010240	012777	000300	015402	MOV	#300,DRPCA	;SELECT CYLINDER 300
2125	010246	112777	000011	015364	MOVB	#11,DRPCS	;ISSUE SEEK TO CYLINDER 300
2126	010254	105777	015360		TSTB	DRPCS	;WAIT FOR DONE
2127	010260	100375			BPL	.-4	
2128	010262	012777	177777	015354	MOV	#-1,DRPWC	;SETUP WORD COUNT
2129	010270	112777	000003	015342	MOVB	#3,DRPCS	;ISSUE A WRITE WHILE HEADS ARE MOVING
2130	010276	105777	015336		TSTB	DRPCS	;WAIT FOR DONE
2131	010302	100375			BPL	.-4	
2132	010304	005777	015330		TST	DRPCS	;ANY ERRORS?
2133	010310	100001			BPL	2\$	;BRANCH IF NO
2134	010312	104400			HLT		;ERROR FOUND AFTER ISSUING A WRITE
2135							;COMMAND WHILE THE HEADS ARE STILL
2136							;MOVING AFTER A SEEK.
2137	010314	032777	004000	015336	2\$:	BIT	#B11,DRPDS
2138	010322	001412			BEQ	3\$	;SEEK INCOMPLETE ERROR?
2139	010324	112777	000015	015306	MOVB	#15,DRPCS	;BRANCH IF NO
2140	010332	105777	015302		TSTB	DRPCS	;ISSUE A HOME COMMAND
2141	010336	100375			BPL	.-4	;WAIT FOR DONE
2142	010340	032777	100000	015312	4\$:	BIT	#B15,DRPDS
2143	010346	001774			BEQ	4\$	
2144	010350	012767	010172	170424	3\$:	MOV	#SKTST,LAD
2145	010356	104000			SCOPE		
2146							
2147							;CHECK THE ABILITY OF CLEAR TO TERMINATE AN OPERATION
2148							;AND SET READY.
2149							
2150	010360	004567	006252		CLRTST:	JSR	R5,DSKNOS
2151	010364	005067	015310		CLR	DMA	;SELECT THE UNIT
2152	010370	005067	015302		CLR	CYLINDER	;CLEAR DISK ADDR
2153	010374	012767	000001	015272	MOV	#1,WRDCT	;SETUP WORD COUNT
2154	010402	012767	025762	015276	MOV	#OUTBUF,BUF	;SETUP BUFFER ADDR
2155	010410	004567	004566		JSR	R5,FUNCT	;ISSUE A WRITE COMMAND
2156	010414	000003			.WORD	3	
2157	010416	012700	000100		MOV	#100,RO	
2158	010422	005300			1\$:	DEC	RO
2159	010424	001376			BNE	1\$	;WAIT A WHILE THEN ISSUE A
2160	010426	112777	000001	015204	MOVB	#1,DRPCS	;CLEAR COMMAND
2161	010434	105777	015200		TSTB	DRPCS	;CLEAR THE CONTROLLER
2162	010440	100401			BMI	2\$	;IS READY SET?
2163	010442	104400			HLT		;BRANCH IF YES
2164							;READY DID NOT SET AFTER ISSUING
2165	010444	000005			2\$:	RESET	;A CLEAR COMMAND DURING A WRITE.
2166	010446	032767	000100	015210	BIT	#B6,FLAG	;TEST FOR MEM. MGR.
2167	010454	001002			BNE	3\$	;BRANCH MEM. MGR IS OFF
2168	010456	005237	177572		INC	#SRO	;TURN ON MEM. MGR.
2169	010462	012767	010360	170312	3\$:	MOV	#CLRTST,LAD
2170	010470	104000			SCOPE		
2171	010472	016767	015166	015234	MOV	FLAG,WORK	;GET UNIT NUMBER
2172	010500	000241			CLC		
2173	010502	006067	015226		ROR	WORK	

D04

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 41
 DZRPBE.P11 29-APR-77 14:41 ***** TEST 3 *****

2174	010506	006067	015222	
2175	010512	000367	015216	
2176	010516	042767	174377	015210
2177	010524	016777	015204	015106
2178	010532	005777	015122	
2179	010535	100375		
2180	010540	032737	002000	177570
2181	010546	001402		
2182	010550	000167	175520	

ROR	WORK
SWAB	WORK
BIC	#174377, WORK
MOV	WORK, @RPCS
TST	@RPDS
BPL	-4
BIT	#810, @SWR
BEG	MEMTST
JMP	RWRCK

;JUSTIFY UNIT NUMBER
 ;CLEAR UNWANTED BITS
 ;LOAD UNIT NUMBER
 ;WAIT FOR UNIT READY

;LOOP ON TEST?
 ;BRANCH IF NO

.SBTTL ***** TEST 4 *****

:THIS ROUTINE CONSIST OF TWO SEGMENTS. THE FIRST
 :PART TEST THE ACCESSIBILITY OF MEMORY WITHOUT
 :UTILIZING MEMORY MANAGEMENT. EACH LOCATION
 :FROM THE END OF THE PROGRAM TO THE TOP OF MEMORY
 : (NOT TO EXCEED 28K) IS WRITTEN WITH ITS ADDR. THIS
 :DATA IS THEN WRITTEN ON THE DISK. THE BUFFER IS
 :CLEARED AND THE DATA IS READ BACK AND VERIFIED.
 :IN PART TWO, THE EXTENDED ADDRESS BITS ARE TESTED.

2183										
2184										
2185										
2186										
2187										
2188										
2189										
2190										
2191										
2192										
2193										
2194	010554	012767	000004	015146	MEMTST:	MOV	#4,TESTNO			
2195	010562	004567	170476			JSR	R5,PRINT\$		;PRINT MESSAGE	
2196	010566	023766				MES6				
2197	010570	016767	015134	170764		MOV	TESTNO,TTY			
2198	010576	004767	170554			JSR	PC,PRINT\$		;TYPE LOCATION-SUPPRESS ZEROS	
2199	010602	012767	177700	015110		MOV	#-100,PASSC		;SETUP ITERATION COUNT NO MEM. MANAG.	
2200	010610	032767	000500	015046		BIT	#500,FLAG		;BK WITH MEM. MANAG. OR NO MEM. MANAG.?	
2201	010616	001003				BNE	1\$		;BR IF TRUE	
2202	010620	012767	177750	015072		MOV	#-30,PASSC		;IF MEMORY MANAGEMENT DO ONLY 24 PASSES	
2203	010626	005067	011116		1\$:	CLR	MEX		;CLEAR DRIVE EXTENDED MEMORY BITS	
2204	010632	004767	006052		RMENT:	JSR	PC,INIT		;INITIALIZE	
2205	010636	032767	000100	015020		BIT	#86,FLAG		;MEMORY MANAGEMENT?	
2206	010644	001520				BEQ	RMMEMT		;YES GO DO MEMORY MANAGEMENT TEST	
2207	010646	016700	011074		MEMBK:	MOV	MEMSIZ,R0		;GET TOP OF CORE	
2208	010652	162700	025762			SUB	#OUTBUF,R0		;DETERMINE SIZE OF BUFFER IN BYTES	
2209	010656	000241				CLC				
2210	010660	006000				ROR	R0		;CONVERT TO WORDS	
2211	010662	042700	000001			BIC	#1,R0		;KEEP NUMBER EVEN	
2212	010666	010067	015002			MOV	R0,WRDCT		;SAVE WORD COUNT OF TRANSFER	
2213	010672	012702	025762		7\$:	MOV	#OUTBUF,R2			
2214	010676	012703	025762			MOV	#OUTBUF,R3		;GENERATE A PATTERN SO THAT EACH	
2215	010702	010322			1\$:	MOV	R3,(R2)+		;LOCATION CONTAINS ITS ADDRESS	
2216	010704	005723				TST	(R3)+			
2217	010706	020267	011034			CMP	R2,MEMSIZ		;HAS ENTIRE PATTERN BEEN GENERATED?	
2218	010712	101773				BLOS	1\$		;BRANCH IF NO	
2219	010714	012767	025762	014764		MOV	#OUTBUF,BUF		;SET UP BUFFER ADDR	
2220	010722	005067	014752			CLR	DMA			
2221	010726	005067	014744			CLR	CYLINDER			
2222	010732	004567	004244			JSR	R5,FUNCT		;WRITE ADDRESS PATTERN	
2223	010736	000003			.WORD	3				
2224	010740	105777	014674			TSTB	@RPCS		;WAIT FOR DONE	
2225	010744	100375				BPL	.-4			
2226	010746	005777	014666			TST	@RPCS		;ANY ERRORS?	
2227	010752	100002				BPL	2\$		;BRANCH IF NO	
2228	010754	104400				HLT			;ERROR AFTER WRITING ADDR PATTERN	
2229	010756	000446				BR	3\$			
2230	010760	016700	014710		2\$:	MOV	WRDCT,R0			
2231	010764	012701	025762			MOV	#OUTBUF,R1			
2232	010770	005021			10\$:	CLR	(R1)+		;CLEAR THE BUFFER	
2233	010772	005300				DEC	R0			
2234	010774	001375				BNE	10\$			
2235	010776	004567	004200			JSR	R5,FUNCT		;READ ADDRESS PATTERN	
2236	011002	000005			.WORD	5				
2237	011004	105777	014630			TSTB	@RPCS		;WAIT FOR DONE	
2238	011010	100375				BPL	.-4			

2239	011012	005777	014622		TST	ARPCS		;ANY ERRORS?
2240	011016	100006			BPL	4\$		;BRANCH IF NO
2241	011020	104400			HLT			;ERROR AFTER READING ADDR PATTERN
2242	011022	032777	040000	014610	BIT	#B14,ARPCS		;IS THIS A DATA ERROR?
2243	011030	001401			BEQ	4\$		;BRANCH IF YES
2244	011032	000420			BR	3\$		
2245	011034	012702	025762		4\$: MOV	#OUTBUF,R2		
2246	011040	010203			MOV	R2,R3		;COMPARE THE ADDR PATTERN
2247	011042	020322			6\$: CMP	R3,(R2)+		;IS DATA CORRECT?
2248	011044	001005			BNE	5\$		;BRANCH IF NO
2249	011046	005723			TST	(R3)+		
2250	011050	020267	010672		CMP	R2,MEMSIZ		;IS ENTIRE BUFFER VERIFIED?
2251	011054	101772			6\$: BLOS	6\$		;BRANCH IF NO
2252	011056	000406			BR	3\$		
2253	011060	010367	006740		5\$: MOV	R3,EXP\$		
2254	011064	016267	17777E	006734	MOV	-2(R2),RECS		
2255	011072	104401			HLT	+1		;COMPARE ERROR UTILIZING ADDR PATTERN
2256	011074	012767	010672	167700	3\$: MOV	#7\$,LAD		;SETUP LOOP ADD
2257	011102	104000			SCOPE			
2258	011104	000527			BR	NONEX		;GO DO REST OF TEST
2259								
2260								;THIS PORTION IS FOR MEMORY MANAGEMENT
2261								
2262	011106	032767	000400	014550	RMMEMT: BIT	#B8,FLAG		;BK MACHINE?
2263	011114	001254			BNE	MEMBK		;BR IF YES
2264	011116	004767	010660		JSR	PC,PARINT		;INITIALIZE KIPARS
2265	011122	004767	010744		1\$: JSR	PC,PARINC		;INCREMENT KIPARS
2266	011126	004767	010770		JSR	PC,PARREG		;GENERATE "MEX" "BUF" "WRDCT"
2267	011132	012700	040000		5\$: MOV	#40000,R0		;STARTING ADDRESS OF BUFFER
2268	011136	016701	014532		MOV	WRDCT,R1		;GET BUFFER SIZE
2269	011142	010010			2\$: MOV	R0,(R0)		;MOVE ADDRESS TO LOCATION
2270	011144	005720			TST	(R0)+		;BUMP LOCATION
2271	011146	005301			DEC	R1		;DONE?
2272	011150	001374			BNE	2\$		;BR IF NO
2273	011152	005067	014522		CLR	DMA		
2274	011156	005067	014514		CLR	CYLINDER		
2275	011162	004567	004014		JSR	R5,FUNCT		;WRITER ADDRESS PATTERN
2276	011166	000003			.WORD	3		
2277	011170	105777	014444		TSTB	ARPCS		;WAIT FOR DONE
2278	011174	100375			BPL	-4		
2279	011176	005777	014436		TST	ARPCS		;ANY ERRORS?
2280	011202	100005			BPL	3\$		;BR IF NO
2281	011204	104400			HLT			;ERROR WRITTING ADDRESS PATTERN
2282	011206	012767	011132	167566	MOV	#5\$,LAD		;SETUP LOOP ADDRESS
2283	011214	104000			SCOPE			
2284	011216	012700	040000		3\$: MOV	#40000,R0		;START OF BUFFER
2285	011222	016701	014446		MOV	WRDCT,R1		;BUFFER SIZE
2286	011226	005020			4\$: CLR	(R0)+		;CLEAR BUFFER
2287	011230	005301			DEC	R1		;DONE?
2288	011232	001375			BNE	4\$		;BR IF NO
2289	011234	004567	003742		6\$: JSR	R5,FUNCT		;READ THE ADDRESS PATTERN
2290	011240	000005			.WORD	5		
2291	011242	105777	014372		TSTB	ARPCS		;WAIT FOR DONE
2292	011246	100375			BPL	-4		
2293	011250	005777	014364		TST	ARPCS		;ANY ERRORS
2294	011254	100011			BPL	7\$		;BR IF NO

G04

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 44
 DZRPBE.P11 29-APR-77 14:41 ***** TEST 4 *****

```

2295 011256 104400 HLT ;DISK ERROR
2296 011260 032777 040000 014352 BIT #B14,ARPCS ;DATA ERROR?
2297 011266 001404 BEQ 7$ ;BR IF YES
2298 011270 012767 011234 167504 MOV #6$,LAD ;SETUP LOOP ADDRESS
2299 011276 104000 SCOPE
2300 011300 012700 040000 7$: MOV #40000,R0 ;START OF BUFFER
2301 011304 016701 014364 MOV WRDCT,R1 ;BUFFER SIZE
2302 011310 010002 9$: MOV R0,R2
2303 011312 022002 CMP (R0)+,R2 ;DATA GOOD?
2304 011314 001004 BNE 8$ ;BR IF DATA NO GOOD
2305 011316 005722 TST (R2)+
2306 011320 005301 DEC R1 ;DONE?
2307 011322 001372 BNE 9$ ;BR IF NO
2308 011324 000413 BR 10$ ;CONTINUE THE TST
2309 011326 010267 006472 8$: MOV R2,EXP$
2310 011332 014067 006470 MOV -(R0),REC$
2311 011336 104405 HLT +5 ;DATA COMPARE ERROR
2312 011340 012767 011310 167434 MOV #9$,LAD ;SETUP LOOP ADDRESS
2313 011346 104000 SCOPE
2314 011350 005720 TST (R0)+ ;TEST REST OF BUFFER
2315 011352 000756 BR 9$ ;TEST MORE
2316 011354 032767 001000 014302 10$: BIT #B9,FLAG ;LAST PAGE?
2317 011362 001657 BEQ 1$ ;BR IF NO
2318 ;CHECK THE SETTING OF NON EXISTENT MEMORY FLAG IN RPER.
2319
2320 011364 032767 000100 014272 NONEX: BIT #B6,FLAG ;IS MEMORY MANAGEMENT ON?
2321 011372 001002 BNE 4$ ;BR IF NOT ON
2322 011374 005037 177572 CLR #SR0 ;IF ON TURN OFF
2323 011400 012767 000001 014266 4$: MOV #1,WRDCT ;SETUP WORD COUNT
2324 011406 012767 160000 014272 MOV #160000,BUF ;LOAD ADDR TO FORCE NON EX MEMORY
2325 011414 004567 003562 JSR R5,FUNCT ;ISSUE A WRITE COMMAND AND
2326 011420 000063 .WORD 63 ;EXPECT NON EX MEMORY
2327 011422 105777 014212 TSTB ARPCS ;WAIT FOR READY
2328 011426 100375 BPL -4
2329 011430 032777 000004 014220 BIT #B2,ARPER ;DID NON EX MEMORY SET
2330 011436 001002 BNE 1$ ;BRANCH IF SET
2331 011440 104400 HLT ;NON EX MEMORY DID NOT SET
2332 011442 000413 BR 2$
2333 011444 032777 040000 014166 1$: BIT #B14,ARPCS ;DID HARD ERROR SET ON NON EX MEMORY
2334 011452 001002 BNE 3$ ;BRANCH IF SET
2335 011454 104400 HLT ;HARD ERROR DID NOT SET
2336 011456 000405 BR 2$
2337 011460 032777 100000 014152 3$: BIT #B15,ARPCS ;DID THE ERROR FLAG SET AFTER NON EX MEMORY
2338 011466 001001 BNE 2$ ;BRANCH IF SET
2339 011470 104400 HLT ;ERROR FLAG DID NOT SET
2340 011472 012767 011400 167302 2$: MOV #4$,LAD
2341 011500 104000 SCOPE
2342
2343 ;IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS
2344 ;BITS.
2345
2346 011502 012737 012310 000004 EXTTST: MOV #EXTTRAP,ARERRVEC ;SETUP TIMEOUT TRAP
2347 011510 012737 000340 000006 MOV #PRI7,ARERRVEC+2
2348 011516 005737 177572 TST #SR0 ;IF MEMORY MANAGEMENT IS NOT
2349 ;AVAILABLE THE PROGRAM WILL TRAP
2350 ;AND TRANSFER TO END OF THE TEST

```


H04

 RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 45
 DZAPBE.P11 29-APR-77 14:41 ***** TEST 4 *****

2351	011522	012737	007600	172356	MOV	#7600, 2#KIPAR7	: OPEN I/O REGISTERS
2352	011530	005037	172340		CLR	2#KIPAR0	: FREE FIRST 4K
2353	011534	012737	000200	172342	MOV	#200, 2#KIPAR1	: ENABLE SECOND 4K
2354	011542	012737	004000	172344	MOV	#4000, 2#KIPAR2	
2355	011550	012737	177406	172300	MOV	#400*256.-400+UP+RW, 2#KIPDR0	: SET KIPDR0=RW UP 400 BLOCKS
2356	011556	012737	177406	172302	MOV	#400*256.-400+UP+RW, 2#KIPDR1	: SET KIPDR1=RW UP 400 BLOCKS
2357	011564	012737	177406	172304	MOV	#400*256.-400+UP+RW, 2#KIPDR2	: SET KIPDR2=RW UP 400 BLOCKS
2358	011572	012737	177406	172316	MOV	#400*256.-400+UP+RW, 2#KIPDR7	: SET KIPDR7=RW UP 400 BLOCKS
2359	011600	012702	040000		MOV	#40000, R2	: R2 GETS BASE ADDRESS
2360	011604	012737	000001	177572	MOV	#1, 2#SR0	: TURN ON MEMORY MANAGEMENT
2361	011612	005712			TST	(R2)	: CHECK IF ENOUGH MEMORY EXISTS TO ALLOW
2362							: TESTING OF THE EXTENDED ADDRESS BITS
2363							: IN THE RP11C. IF NOT ENOUGH MEMORY
2364							: EXISTS, A TIME-OUT TRAP WILL OCCUR AND
2365							: TRANSFER CONTROL TO THE END OF THIS
2366							: TEST.
2367	011614	012737	002000	172344	MOV	#2000, 2#KIPAR2	: SET UP FOR THE EXTENDED ADDRESS TESTS
2368	011622	012737	012274	000004	MOV	#EXTRP, 2#ERRVEC	: RESTORE TIME-OUT TRAP VECTOR
2369	011630	012712	177777		MOV	#177777, (R2)	: INSERT PATTERN INTO 200000
2370	011634	012767	000002	014032	MOV	#2, WRDCT	: SETUP WORDCOUNT
2371	011642	005067	010102		CLR	MEX	: CLEAR EXTENDED MEMORY BITS FOR CONTROLLER
2372	011646	012767	177777	014032	MOV	#177777, BUF	: SETUP BUS ADDR
2373	011654	004567	003322		JSR	R5, FUNCT	: WRITE TWO WORDS ON DISK. RPBA
2374	011660	000003					: STARTS AT 177777 TO FORCE CARRY
2375							: TO SET MEX0
2376	011662	105777	013752		TSTB	2#RPCS	: WAIT FOR READY
2377	011666	100375			BPL	.-4	
2378	011670	005777	013744		TST	2#RPCS	
2379	011674	100002			BPL	1\$	
2380	011676	104400			HLT		: STATUS ERROR AFTER 2 WORD WRITE
2381	011700	000447			BR	2\$	: USING MEX0
2382	011702	032777	000020	013730	BIT	#B4, 2#RPCS	: MEX0 SHOULD HAVE SET?
2383	011710	001002			BNE	3\$	: BRANCH IF SET
2384	011712	104400			HLT		: MEX0 DID NOT SET
2385	011714	000441			BR	2\$	
2386	011716	005012			CLR	(R2)	: CLEAR LOCATION 200000
2387	011720	004567	003256		JSR	R5, FUNCT	: READ TWO WORDS INTO LOCATIONS
2388	011724	000005					: 177777 AND 200000.
2389	011726	105777	013706		TSTB	2#RPCS	: WAIT FOR READY
2390	011732	100375			BPL	.-4	
2391	011734	005777	013700		TST	2#RPCS	: ANY ERRORS?
2392	011740	100002			BPL	4\$	: BRANCH IF NO
2393	011742	104400			HLT		: ERROR AFTER READING 2 WORDS
2394	011744	000425			BR	2\$	
2395	011746	032777	000020	013664	BIT	#B4, 2#RPCS	: DID MEX0 SET?
2396	011754	001002			BNE	5\$	: BRANCH IF YES
2397	011756	104400			HLT		: MEX0 DID NOT SET AFTER 2 WORD
2398	011760	000417			BR	2\$	: READ STARTING AT 177777
2399	011762	022712	177777		CMP	#177777, (R2)	: WAS DATA READ INTO LOCATION
2400	011766	001407			BEQ	6\$	: 200000 CORRECTLY? - BRANCH IF YES
2401	011770	012767	177777	006026	MOV	#177777, EXP\$	
2402	011776	011267	006024		MOV	(R2), REC\$	
2403	012002	104401			HLT	+1	: DATA COMPARE ERROR AT 200000
2404	012004	000405			BR	2\$	: IF RECEIVED=0 - LOCATION WASN'T ACCESSED
2405	012006	032777	000040	013624	BIT	#B5, 2#RPCS	: MEX1 SHOULD BE CLEAR
2406	012014	001401			BEQ	2\$	: BRANCH IF CLEAR

2407	012016	104400				HLT		:MEX1 IS SET - SHOULD NOT BE
2408	012020	012767	011630	166754	2\$:	MOV	#7\$,LAD	:SETUP ERROR LOOP
2409	012026	104000				SCOPE		
2410	012030	012737	004000	172344	EXTT1:	MOV	#4000, @#KIPAR2	
2411	012036	012702	040000			MOV	#40000, R2	:R2 EQUALS THE BASE ADDR
2412	012042	012712	177777		7\$:	MOV	#177777, (R2)	:INSERT PATTERN INTO 400000
2413	012046	012767	177777	013632		MOV	#177777, BUF	:SETUP BUS ADDR
2414	012054	004567	003122			JSR	R5, FUNCT	
2415	012060	000023			.WORD	23		
2416	012062	105777	013552			TSTB	@RPCS	:WAIT FOR READY
2417	012066	100375				BPL	4	
2418	012070	005777	013544			TST	@RPCS	:ANY ERRORS?
2419	012074	100002				BPL	4\$	:BRANCH IF NO
2420	012076	104400				HLT		
2421	012100	000455				BR	2\$	:ERROR AFTER READING 2 WORDS
2422	012102	032777	000020	013530	4\$:	BIT	#B4, @RPCS	:DID MEX0 CLEAR?
2423	012110	001402				BEQ	5\$	:BRANCH IF YES
2424	012112	104400				HLT		:MEX0 DID NOT CLEAR AFTER 2 WORD
2425	012114	000447				BR	2\$	:READ STARTING AT 377777
2426	012116	032777	000040	013514	5\$:	BIT	#B5, @RPCS	:DID MEX1 SET?
2427	012124	001002				BNE	10\$	:BRANCH IF YES
2428	012126	104400				HLT		:MEX1 DI NOT SET WITH A TWO WORD
2429								:TRANSFER STARTING AT 377777
2430	012130	000441				BR	2\$	
2431	012132	005012			10\$:	CLR	(R2)	:CLEAR LOCATION 400000
2432	012134	004567	003042			JSR	R5, FUNCT	:READ TWO WORDS STARTING AT 377777
2433	012140	000025			.WORD	25		
2434	012142	105777	013472			TSTB	@RPCS	:WAIT FOR READY
2435	012146	100375				BPL	4	
2436	012150	005777	013464			TST	@RPCS	:ANY ERRORS?
2437	012154	100002				BPL	11\$	:BRANCH IF NO
2438	012156	104400				HLT		:ERROR WHILE READING TWO WORDS
2439	012160	000425				BR	2\$	
2440	012162	032777	000020	013450	11\$:	BIT	#B4, @RPCS	:DID MEX0 CLEAR?
2441	012170	001402				BEQ	12\$	:BRANCH IF YES
2442	012172	104400				HLT		:MEX0 DID NOT CLEAR AFTER 2 WORD
2443	012174	000417				BR	2\$	:READ STARTING AT 377777
2444	012176	022712	177777		12\$:	CMP	#177777, (R2)	:WAS DATA READ INTO LOCATION 400000
2445	012202	001407				BEQ	6\$	:CORRECTLY? - BRANCH IF YES
2446	012204	012767	177777	005612		MOV	#177777, EXP\$	
2447	012212	011267	005610			MOV	(R2), REC\$	
2448	012216	104401				HLT	+1	:DATA COMPARE ERROR AT 400000 IF
2449	012220	000405				BR	2\$	:RECEIVED=0 - LOCATION WASN'T ACCESSED
2450	012222	032777	000040	013410	6\$:	BIT	#B5, @RPCS	:DID MEX1 SET?
2451	012230	001001				BNE	2\$	
2452	012232	104400				HLT		:MEX1 DID NOT SET AFTER 2 WORD TRANSFER
2453								:STARTING AT 377777
2454	012234	012767	012042	166540	2\$:	MOV	#7\$,LAD	:SETUP ERROR LOOP
2455	012242	104000				SCOPE		
2456	012244	000413				BR	EXTRP	:CLEAR MEMORY MANAGEMENT
2457	012246	005267	013446		EXTEND:	INC	PASSC	:INCREMENT ITERATION COUNT
2458	012252	001402				BEQ	1\$	
2459	012254	000167	176352			JMP	RMENT	
2460	012260	032737	002000	177570	1\$:	BIT	#B10, @#SWR	:LOOP ON TEST?
2461	012266	001420				BEQ	DATA	
2462	012270	000167	176336			JMP	RMENT	


```

2463
2464 012274 032767 000100 013362 EXTRP: BIT      #B6,FLAG      ;USING MEMORY MANAGEMENT?
2465 012302 001402          BEQ      EXTRP      ;BR IF YES
2466 012304 005037 177572          CLR      #SRO      ;NO TURN IT OFF
2467 012310 012706 000500          EXTTRP: MOV     #STKPTR,SP ;RESTORE STACK
2468 012314 012767 000006 165462 MOV     #6,4
2469 012322 005067 165460          CLR      6
2470 012326 000747          BR       EXTEND
2471          .SBTTL ***** TEST 5 *****
2472
2473 ;WRITE, WRITE CHECK, AND READ OPERATIONS ARE PERFORMED ON THE DRIVE
2474 ;THE DATA IS FIRST WRITTEN AND THEN WRITE CHECKED. THEN THE DATA
2475 ;IS READ. IF THE DATA IS TO BE COMPARED, THE INPUT BUFFER IS CLEARED
2476 ;RIGHT AFTER READ IS ISSUED. THEN THE DATA IS COMPARED WHILE READ
2477 ;IS IN PROGRESS. THIS IS DONE TO IMPROVE EFFICIENCY. THIS SEQUENCE
2478 ;IS REPEATED FOR THE ENTIRE PACK SURFACE FOR EACH OF THE 22 PATTERNS.
2479 ;ERRORS OCCURING USING RANDOM SEEKS DURING THE WRITE CHECK OR ARE
2480 ;RECOVERABLE AFTER FIRST RETRY DURING READ OR ARE ELIMINATED BY
2481 ;DISABLING RANDOM SEEKS ARE PROBABLY DUE TO VIBRATING HEADS
2482 ;AFTER A SEEK.
2483
2484 012330 012767 000005 013372 DATAT: MOV     #5,TESTNO
2485 012336 004567 166722          JSR      R5,PRINT$      ;PRINT MESSAGE
2486 012342 023766          MES6
2487 012344 016767 013360 167210 MOV     TESTNO,TTY
2488 012352 004767 167000          JSR      PC,PRINT$      ;TYPE LOCATION-SUPPRESS ZEROS
2489 012356 005067 013372          CLR      CNTA      ;INITIALIZE READ COUNTER
2490 012362 005067 007362          CLR      MEX      ;CLEAR EXTENDED MEMORY BITS IN CONTROLLER
2491 012366 016700 013312          RDATAT: MOV    PATNU,RO      ;GET PATTERN NO.
2492 012372 000241          CLC
2493 012374 006000          ROR
2494 012376 010037 177570          MOV     RO,#SWR
2495 012402 005067 013270          CLR      CYLINDER      ;DISPLAY PATTERN NO. IN USE
2496 012406 005067 013266          CLR      DMA
2497 012412 012737 000200 177776 MOV     #PRI4,#PSW      ;ENABLE INT SYSTEM
2498 012420 016767 013264 013246 MOV     SWRDCT,WRDCT
2499 012426 012777 015272 013200 MOV     #DKINT,#VECTOR ;SETUP DISK VECTOR
2500 012434 012777 000340 013174 MOV     #340,#STATUS
2501 012442 032767 000500 013214 BIT      #500,FLAG      ;BK WITH MEM. MANAG. OR NO MEM. MANAG
2502 012450 001006          BNE      DATP      ;BR IF TRUE
2503 012452 004767 007324          JSR      PC,PARINT      ;INITIALIZE KIPARS
2504 012456 000240          NOP
2505 012460 000240          NOP
2506 012462 004767 007434          JSR      PC,PARREG
2507 012466 004767 004414          JSR      PC,PASEL
2508 012472 032767 000500 013164 DATP: BIT      #500,FLAG      ;NEEDED FOR CONFORMITY
2509 012500 001403          BEQ      1$      ;SETUP "MEX","WRDCT","BUF"
2510 012502 012767 025762 013176 MOV     #OUTBUF,BUF      ;GENERATE PATTERN
2511 012510 032767 040000 013146 1$: BIT      #B14,FLAG      ;BK MEM. MANAG. OR NO MEM. MANAG?
2512 012516 001426          BEQ      WRICK      ;BR IF NOT TRUE
2513 012520 004767 003444          LDAT: JSR     PC,OPDSEL      ;SETUP BUFFER ADDR
2514 012524 004767 003136          JSR      PC,SEEK      ;WRITE?
2515 012530 004567 002446          JSR      R5,FUNCT      ;BRANCH IF NO
2516 012534 000103          .WORD 103      ;ANY OPERATOR ADDR PARAMETERS?
2517 012536 032737 010000 177570 BIT      #B12,#SWR      ;GO DO A RANDOM SEEK
2518 012544 001003          BNE      1$      ;WRITE WITH INTERRUPTS
          ;DETERMINE HOW TO WAIT FOR INT

```


K04

 PUBLIC RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 48
 DZAPBE.PYI 29-APR-77 14:41 ***** TEST 5 *****

2519	012546	004767	007514			JSR	PC,NPR		;GENERATE WORSE CASE NPR CYCLES
2520	012552	000401				BR	2\$		
2521	012554	000001			1\$:	WAIT			
2522	012556	012767	012520	166216	2\$:	MOV	#LDAT,LAD		;SETUP LOOP ADDR
2523	012564	104000				SCOPE			
2524	012566	004767	003434			JSR	PC,DISBUF		;PREPARE NEW DISK ADDR
2525	012572	000752				BR	LDAT		
2526	012574	032767	020000	013062	WRICK:	BIT	#B13,FLAG		;WRITE CHECK?
2527	012602	001424				BEQ	DREAD		;BRANCH IF NO
2528	012604	004767	003360		3\$:	JSR	PC,OPDSEL		;ANY OPERATOR ADDR PARAMETERS?
2529	012610	004567	002366			JSR	RS,FUNCT		;WRITE CHECK THE DATA
2530	012614	000107			.WORD	107			
2531	012616	032737	010000	177570		BIT	#B12,2\$SWR		;DETERMINE HOW TO WAIT FOR INT
2532	012624	001003				BNE	1\$		
2533	012626	004767	007434			JSR	PC,NPR		;GENERATE WORSE CASE NPR CYCLES
2534	012632	000401				BR	2\$		
2535	012634	000001			1\$:	WAIT			
2536	012636	012767	012604	166136	2\$:	MOV	#3\$,LAD		;SETUP LOOP ADDR
2537	012644	104000				SCOPE			
2538	012646	004767	003354			JSR	PC,DISBUF		;PREPARE NEW DISK ADDR
2539	012652	000754				BR	3\$		
2540	012654	032767	010000	013002	DREAD:	BIT	#B12,FLAG		;READ?
2541	012662	001002				BNE	1\$		;BRANCH IF YES
2542	012664	000167	000444			JMP	MSTR		;GET AROUND READ
2543	012670	032767	000500	012766	1\$:	BIT	#500,FLAG		;BK WITH MEM MANAG. OR NO MEM MANAG.
2544	012676	001006				BNE	READ1		;BR IF TRUE
2545	012700	004767	007076			JSR	PC,PARINT		;INITIALIZE KIPARS NEEDED
2546	012704	000240				NOP			;FOR CONFORMITY
2547	012706	000240				NOP			
2548	012710	004767	007206			JSR	PC,PARREG		;SETUP "BUF",WRDCT", "MEX"
2549	012714	005067	013034		READ1:	CLR	CNTA		;INITIALIZE READ COUNTER
2550	012720	005067	013000		ESH:	CLR	RDERR		;CLEAR READ ERROR COUNT
2551	012724	004767	003240			JSR	PC,OPDSEL		;ANY OPERATOR ADDR PARAMETERS?
2552	012730	004767	002732			JSR	PC,SEEK		;GO DO A RANDOM SEEK
2553	012734	005067	171552		DSKRD:	CLR	INTFLG		;CLEAR THE INTERRUPT FLAG
2554	012740	004567	002236			JSR	RS,FUNCT		;READ THE DATA
2555	012744	000105			.WORD	105			
2556	012746	032737	001000	177570		BIT	#B9,2\$SWR		;COMPARE DATA?
2557	012754	001411				BEQ	1\$		;BRANCH IF YES
2558	012756	032737	010000	177570		BIT	#B12,2\$SWR		;DETERMINE HOW TO WAIT FOR INT
2559	012764	001003				BNE	2\$		
2560	012766	004767	007274			JSR	PC,NPR		;GENERATE WORSE CASE NPR CYCLES
2561	012772	000447				BR	3\$		
2562	012774	000001			2\$:	WAIT			
2563	012776	000445				BR	3\$		
2564	013000	016700	012670		1\$:	MOV	WRDCT,R0		;CLEAR THE INPUT BUFFER
2565	013004	012702	000020			MOV	#16,R2		
2566	013010	012701	025762			MOV	#OUTBUF,R1		
2567	013014	032767	000500	012642		BIT	#500,FLAG		;BK WITH MEM. MANAG. OR NO MEM. MANAG.
2568	013022	001002				BNE	4\$		;BR IF TRUE
2569	013024	012701	040000			MOV	#40000,R1		;START OF MEM. MANAG. OUT BUFF
2570	013030	005700			4\$:	TST	R0		;TEST WORD COUNT IS IT AT ZERO?
2571	013032	100422				BMI	10\$		;BR IF CLEAR IS FINISHED
2572	013034	005021				CLR	(R1)+		
2573	013036	005021				CLR	(R1)+		
2574	013040	005021				CLR	(R1)+		

L04

2575	013042	005021			CLR	(R1)+	
2576	013044	005021			CLR	(R1)+	
2577	013046	005021			CLR	(R1)+	
2578	013050	005021			CLR	(R1)+	
2579	013052	005021			CLR	(R1)+	
2580	013054	005021			CLR	(R1)+	
2581	013056	005021			CLR	(R1)+	
2582	013060	005021			CLR	(R1)+	
2583	013062	005021			CLR	(R1)+	
2584	013064	005021			CLR	(R1)+	
2585	013066	005021			CLR	(R1)+	
2586	013070	005021			CLR	(R1)+	
2587	013072	005021			CLR	(R1)+	
2588	013074	160200			SUB	R2,R0	;DECREMENT WORD COUNT
2589	013076	001354			BNE	4\$	
2590	013100	004767	004760	10\$:	JSR	PC,COMPAR	;COMPARE THE DATA
2591	013104	105777	012530		TSTB	ARPCS	;WAIT FOR READY
2592	013110	100375			BPL	-4	
2593	013112	005767	166122	3\$:	TST	ERRFLG	;WERE THERE ANY ERRORS
2594	013116	001424			BEQ	5\$	;BRANCH IF NO
2595	013120	005267	012600		INC	RDERR	;UPDATE ERROR COUNT
2596	013124	022767	000024	012572	CMP	#20.,RDERR	;MORE THAN 20 ERRORS?
2597	013132	001416			BEQ	5\$	;BRANCH IF YES
2598	013134	022767	000012	012562	CMP	#10.,RDERR	;IS THIS TENTH ERROR?
2599	013142	001274			BNE	DSKRD	;BRANCH IF NO
2600	013144	112777	000015	012466	MOVB	#15,ARPCS	;HOME THE HEADS
2601	013152	105777	012462		TSTB	ARPCS	;WAIT FOR DONE
2602	013156	100375			BPL	-4	
2603	013160	005777	012474		TST	ARPCS	
2604	013164	100375			BPL	-4	;WAIT FOR READY
2605	013166	000662			BR	DSKRD	
2606	013170	005767	012530	5\$:	TST	RDERR	
2607	013174	001417			BEQ	6\$	
2608	013176	022767	000013	012520	CMP	#11.,RDERR	;WAS READ ERROR STILL THERE AFTER A RECALL
2609	013204	002003			BGE	9\$	;BR IF NO
2610	013206	012767	000002	012540	MOV	#2,CNTA	;YES DON'T TRY TO READ AGAIN UNLESS LOOPING
2611	013214			9\$:			
2612	013214	004567	166044		JSR	R5,PRINT\$	;PRINT MESSAGE
2613	013220	024001			MES7		
2614	013222	016767	012476	166332	MOV	RDERR,TTY	
2615	013230	004767	166122		JSR	PC,PRINTS	;TYPE LOCATION-SUPPRESS ZEROS
2616	013234	005067	012464	6\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
2617	013240	012767	012720	165534	MOV	#ESH,LAD	;LOOP ADDR
2618	013246	104000			SCOPE		
2619	013250	005267	012500		INC	CNTA	;INCREMENT READ COUNTER
2620	013254	022767	000003	012472	CMP	#3,CNTA	;DONE 3 READS?
2621	013262	001216			BNE	ESH	;BR IF NO
2622	013264	032767	000500	012372	BIT	#500,FLAG	;BK MEM. MANAG. OR NO MEM. MANAG.
2623	013272	001012			BNE	8\$	;BR IF TRUE
2624	013274	032767	001000	012362	BIT	#89,FLAG	;LAST PAGE TRANSFERED?
2625	013302	001402			BEQ	7\$	;BR IF NO
2626	013304	004767	006472		JSR	PC,PARINT	;GO REINITIALIZE KIPARS
2627	013310	004767	006556	7\$:	JSR	PC,PARINC	;GO INCREMENT KIPARS BY 1K
2628	013314	004767	006602		JSR	PC,PARREG	;GO SETUP "MEX","WRDCT","BUF"
2629	013320	004767	002702	8\$:	JSR	PC,DISBUF	;GET NEW DISK ADDR
2630	013324	000401			BR	11\$	

MO4

2631	013326	000402			BR	MSTR	
2632	013330	000167	177360		11\$: JMP	READ1	
2633	013334	032767	000040	012322	MSTR: BIT	#B5,FLAG	;LOOPING ON AN OPERATOR ADDR?
2634	013342	001402			BEQ	1\$	;NO CONTINUE
2635	013344	000167	177116		JMP	DATP	;YES
2636	013350	005767	012310		1\$: TST	FLAG	
2637	013354	100002			BPL	2\$	;UNDER PROGRAM CONTROL
2638	013356	000167	001034		JMP	MULCHK	;OPERATOR SELECTED PATTERN
2639	013362	062767	000002	012314	2\$: ADD	#2,PATNU	;INC PATTERN INDEX
2640	013370	022767	000036	012306	CMP	#36,PATNU	;PATTERNS EXCEEDED?
2641	013376	001402			BEQ	3\$	
2642	013400	000167	176762		JMP	RDATAT	;NOT YET
2643	013404	005067	012274		3\$: CLR	PATNU	;LAST PATTERN USED
2644	013410	032737	002000	177570	BIT	#B10,#SWR	;LOOP ON TEST?
2645	013416	001402			BEQ	RANEX	;NO..GO TO RANDOM TEST
2646	013420	000167	176742		JMP	RDATAT	;YES
2647					.SBTTL	***** TEST 6 *****	
2648							
2649							
2650							;THIS IS A RANDOM ADDRESS AND DATA TEST.
2651							;ERRORS OCCURING WHICH ARE RECOVERABLE AFTER THE FIRST RETRY ARE PROBABLY
2652							;DUE TO VIBRATING HEADS AFTER A SEEK, TRY DISABLEING RANDOM SEEKS.
2653	013424	012767	000006	012276	RANEX: MOV	#6,TESTNO	
2654	013432	004567	165626		JSR	RS,PRINT\$	;PRINT MESSAGE
2655	013436	023766			MES6		
2656	013440	016767	012264	166114	MOV	TESTNO,TTY	
2657	013446	004767	165704		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2658	013452	005067	006272		CLR	MEX	;CLEAR EXTENDED MEMORY BITS IN CONTROLER
2659	013456	004767	003226		RRANEX: JSR	PC,INIT	
2660	013462	032767	000500	012174	BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM. MANAG.
2661	013470	001004			BNE	1\$	;BR IF TRUE
2662	013472	004767	006304		JSR	PC,PARINT	;INITALIZE ALL KIPARS
2663	013476	000240			NOP		;NEEDED FOR CONFORMITY
2664	013500	000240			NOP		
2665	013502	012767	000034	012174	1\$: MOV	#34,PATNU	
2666	013510	012767	175300	012202	MOV	#-2500,PASSC	;SET UP PASS COUNT
2667	013516	012737	000200	177776	MOV	#PRI4,#PSW	
2668	013524	012767	000700	012142	WRLG: MOV	#700,WRDCT	;SET UP WORD COUNT TO 1+ SECTOR
2669	013532	016767	012136	012174	MOV	WRDCT,WORK	
2670	013540	012701	025762		MOV	#OUTBUF,R1	
2671	013544	032767	000500	012112	BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
2672	013552	001002			BNE	4\$	;BR IF TRUE
2673	013554	012701	040000		MOV	#40000,R1	;START OF OUTBUFF WITH MEMORY MANAG.
2674	013560	004767	003430		4\$: JSR	PC,RANDOM	;GENERATE RANDOM PATTERN
2675	013564				1\$:		
2676	013564	004767	165774		JSR	PC,RAND\$	;GENERATE TWO RANDOM NOS.
2677	013570	016767	166106	012136	MOV	LONUM,WORK	
2678	013576	016767	166076	012132	MOV	HINUM,WORK1	
2679	013604	042767	177000	012122	BIC	#177000,WORK	
2680	013612	022767	000625	012114	CMP	#625,WORK	;FORM RANDOM CYL ADDR
2681	013620	002761			BLT	1\$	
2682	013622	016767	012106	012046	MOV	WORK,CYLINDER	;SAVE IT
2683	013630	042767	160360	012100	BIC	#160360,WORK1	
2684	013636	122767	000010	012072	CMPB	#10,WORK1	;FORM RANDOM SECTOR ADDR
2685	013644	101003			BHI	2\$	
2686	013646	042767	000010	012062	BIC	#10,WORK1	

2687	013654	122767	000023	012055	2\$:	CMPB	#23,WORK1+1	
2688	013662	101003				BHI	3\$	;FORM RANDOM HEAD ADDR
2689	013664	142767	000014	012045		BICB	#14,WORK1+1	
2690	013672	016767	012040	012000	3\$:	MOV	WORK1,DMA	;SAVE DESK ADDR.
2691	013700	012767	025762	012000	RANLOP:	MOV	#OUTBUF,BUF	;SETUP OUTPUT BUFFER
2692	013706	032767	000500	011750		BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
2693	013714	001003				BNE	13\$	;BR IF TRUE
2694	013716	012767	040000	011762		MOV	#40000,BUF	;START OF OUTBUFFER WITH MEMORY MANAG.
2695	013724	004767	001736		13\$:	JSR	PC,SEEK	;DO A RANDOM SEEK
2696	013730	004567	001246			JSR	R5,FUNCT	;WRITE RANDOM DATA AND
2697	013734	000103			.WORD	103		;ENABLE INTERRUPTS
2698	013736	032737	010000	177570		BIT	#B12,2#SWR	;DETERMINE HOW TO WAIT FOR INT
2699	013744	001003				BNE	2\$	
2700	013746	004767	006314			JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2701	013752	000401				BR	4\$	
2702	013754	000001			2\$:	WAIT		
2703	013756	012767	013700	165016	4\$:	MOV	#RANLOP,LAD	;SETUP LOOP ADDR
2704	013764	104000				SCOPE		
2705	013766	004567	001210		7\$:	JSR	R5,FUNCT	;WRITE CHECK THE DATA AND
2706	013772	000107			.WORD	107		;ENABLE INTERRUPT
2707	013774	032737	010000	177570		BIT	#B12,2#SWR	;HOW TO WAIT FOR INT?
2708	014002	001003				BNE	1\$	
2709	014004	004767	006256			JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2710	014010	000401				BR	5\$	
2711	014012	000001			1\$:	WAIT		
2712	014014	012767	013766	164760	5\$:	MOV	#7\$,LAD	;SETUP LOOP ADDR
2713	014022	104000				SCOPE		
2714	014024	005067	011724			CLR	CNTA	;INITIALIZE READ COUNTER
2715	014030	005067	011670		8\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
2716	014034	004767	001626			JSR	PC,SEEK	;DO A RANDOM SEEK
2717	014040	005067	170446		11\$:	CLR	INTFLG	
2718	014044	004567	001132			JSR	R5,FUNCT	;READ RANDOM DATA AND
2719	014050	000105			.WORD	105		;ENABLE INTERRUPT
2720	014052	012700	000700			MOV	#700,R0	;SET UP TO CLEAR INPUT BUFFER
2721	014056	012702	000020			MOV	#16.,R2	
2722	014062	012701	025762			MOV	#OUTBUF,R1	;START OF INPUT BUFFER
2723	014066	032767	000500	011570		BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM MANAG?
2724	014074	001002				BNE	15\$	;BR IF TRUE
2725	014076	012701	040000			MOV	#40000,R1	;START OF OUTBUF WITH MEM. MANAG.
2726	014102				15\$:			
2727	014102	005021				CLR	(R1)+	
2728	014104	005021				CLR	(R1)+	
2729	014106	005021				CLR	(R1)+	
2730	014110	005021				CLR	(R1)+	
2731	014112	005021				CLR	(R1)+	
2732	014114	005021				CLR	(R1)+	
2733	014116	005021				CLR	(R1)+	
2734	014120	005021				CLR	(R1)+	
2735	014122	005021				CLR	(R1)+	
2736	014124	005021				CLR	(R1)+	
2737	014126	005021				CLR	(R1)+	
2738	014130	005021				CLR	(R1)+	
2739	014132	005021				CLR	(R1)+	
2740	014134	005021				CLR	(R1)+	
2741	014136	005021				CLR	(R1)+	
2742	014140	005021				CLR	(R1)+	

2743	014142	160200			SUB	R2,PC	;DECREMENT THE WORD COUNT
2744	014144	001356			BNE	15\$	;GO CLR MORE
2745	014146	032737	010000	177570	BIT	#B12,2#SWR	;HOW TO WAIT FOR INT?
2746	014154	001003			BNE	3\$	
2747	014156	004767	006104		JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2748	014162	000401			BR	6\$	
2749	014164	000001			WAIT		
2750	014166	032737	001000	177570	3\$: BIT	#B9,2#SWR	;COMPARE FOR ERRORS?
2751	014174	001006			6\$: BNE	9\$	;BRANCH IF NO
2752	014176	032777	040000	011434	BIT	#B14,2RPCS	;HARD ERROR?
2753	014204	001002			BNE	9\$	;BRANCH IF YES
2754	014206	004767	003652		JSR	PC,COMPARE	;COMPARE DATA FOR ERRORS
2755	014212	005767	165022		9\$: TST	ERRFLG	;READ ERROR?
2756	014216	001424			BEQ	10\$	;BRANCH IF NO
2757	014220	005267	011500		INC	RDERR	;UPDATE ERROR COUNT
2758	014224	022767	000024	011472	CMP	#20.,RDERR	;20 ERRORS YET?
2759	014232	001416			BEQ	10\$	;BRANCH IF YES
2760	014234	022767	000012	011462	CMP	#10.,RDERR	;IS THIS TENTH ERROR?
2761	014242	001276			BNE	11\$	;BRANCH IF NO
2762	014244	112777	000015	011366	MOVB	#15,2RPCS	;ISSUE HOME COMMAND
2763	014252	105777	011362		TSTB	2RPCS	;WAIT FOR DONE
2764	014256	100375			BPL	-4	
2765	014260	005777	011374		TST	2RPDS	;WAIT FOR READT
2766	014264	100375			BPL	-4	
2767	014266	000664			BR	11\$	
2768	014270	005767	011430		10\$: TST	RDERR	
2769	014274	001417			BEQ	21\$	
2770	014276	022767	000013	011420	CMP	#11.,RDERR	;DID RECALL HELP RECOVERY
2771	014304	002003			BGE	14\$	;BR IF YES
2772	014306	012767	000011	011440	MOV	#9.,CNTA	;NO, DON'T TRY READING AGAIN
2773	014314				14\$: JSR	R5,PRINT\$	;PRINT MESSAGE
2774	014314	004567	164744		MES7		
2775	014320	024001			MOV	RDERR,TTY	
2776	014322	016767	011376	165232	JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2777	014330	004767	165022		CLR	RDERR	;CLEAR READ ERROR COUNTER
2778	014334	005067	011364		21\$: MOV	#8\$,LAD	;SET UP LOOP ADDR
2779	014340	012767	014030	164434	SCOPE		
2780	014346	104000			INC	CNTA	;INCREMENT READ COUNTER
2781	014350	005267	011400		CMP	#10.,CNTA	;DONE 10 DISK READS?
2782	014354	022767	000012	011372	BNE	8\$	;BR IF NO
2783	014362	001222			INC	PASSC	;INCREMENT PASS COUNT
2784	014364	005267	011330		BEQ	12\$	;BRANCH IF DONE
2785	014370	001402			JMP	WRLG	;CONTINUE
2786	014372	000167	177126		12\$: CLR	PATNU	
2787	014376	005067	011302		BIT	#B10,2#SWR	;LOOP ON TEST?
2788	014402	032737	002000	177570	BEQ	MULCHK	;NO
2789	014410	001402			JMP	RRANEX	;LOOP
2790	014412	000167	177040				
2791							
2792							
2793							
2794							
2795							
2796							
2797							
2798	014416	005067	011256		MULCHK: CLR	DMA	

;CHECK FOR MULTI DISK MODE
 ;IF IN MULTI DISK MODE REPORT "END"
 ;IF LAST DISK ON SYSTEM HAS BEEN
 ;EXERCISED.

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 53
 DZRPBE.P11 29-APR-77 14:41 ***** TEST 6 *****

2799	014422	005067	011250		CLR	CYLINDER	; CLEAR ADDRESS REGISTERS
2800	014426	032767	004000	011230	BIT	#B11, FLAG	; ARE WE IN MULTI DISK MODE
2801	014434	001422			BEQ	REPOEN	; REPORT "END"
2802	014436	016767	011222	011270	MOV	FLAG, WORK	; WHAT DISK ARE WE ON
2803	014444	042767	177743	011262	BIC	#177743, WORK	; IF LAST DISK ON SYSTEM
2804	014452	026767	011256	011234	CMP	WORK, DSKNOR	; REPORT END
2805	014460	001004			BNE	INDRVE	
2806	014462	042767	000034	011174	BIC	#34, FLAG	
2807	014470	000404			BR	REPOEN	; REPORT "END" LAST DISK
2808	014472	062767	000004	011164	INDRVE: ADD	#4, FLAG	; INC. DISK NO.
2809	014500	000426			BR	EXTPP	; EXERCISE DISK
2810	014502	005267	164272		REPOEN: INC	ICNT	; INCREMENT PASS COUNTER
2811	014506	004567	164552		JSR	R5, PRINTS	; PRINT MESSAGE
2812	014512	023576			MES1		; REPORT END OF PASS
2813	014514	016767	164260	165040	MOV	ICNT, TTY	
2814	014522	004767	164630		JSR	PC, PRINTS	; TYPE LOCATION-SUPRESS ZEROS
2815	014526	013701	000042		MOV	#42, R1	; GET MONITOR RETURN ADDRESS
2816	014532	001411			BEQ	EXTPP	; BRANCH IF NOT UNDER MONITOR
2817	014534	032767	000100	011122	BIT	#B6, FLAG	; IS MEM. MGR. ON
2818	014542	001401			BEQ	MEXIT	; YES , BY PASS RESET
2819	014544	000005			RESET		
2820	014546	004711			MEXIT: JSR	PC, (R1)	; EXIT TO THE MONITOR
2821	014550	000240			NOP		
2822	014552	000240			NOP		
2823	014554	000240			NOP		
2824	014556	000167	166776		EXTPP: JMP	ADTST	; RECYCLE
2825							
2826							

.SBTTL ***** TEST 7 *****

;TEST THE ABILITY OF THE RP11C TO SENSE POWER FAILURE
 ;AND TO HOME THE HEADS. WHEN POWER IS RESTORED
 ;THE CYLINDER ADDRESS IS TESTED FOR ZERO. AFTER TYPING THE MESSAGE
 ;REQUESTING POWER TO BE TURNED OFF THE PROGRAM GOES INTO
 ;A LOOP READING FROM THE DISK. AFTER POWER IS RESTORED,
 ;MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT PUT ANY
 ;JUNK INTO MEMORY WHILE POWER WAS GOING DOWN.

2827					PFTST:	MOV	#STKPTR,SP	
2828						MOV	#PFD,24	;SET UP POWER FAIL VECTOR
2829						MOV	#PRI7,26	;LOCKOUT INTERRUPTS
2830						JSR	RS,DSKNOS	;SELECT THE UNIT
2831						MOV	#625,ARPCA	;SELECT CYLINDER 625
2832						BIS	#11,ARPCS	;ISSUE SEEK COMMAND
2833						MOV	#OUTBUF,RO	
2834					1\$:	MOV	#25252,(RO)+	;FILL MEMORY WITH CHECKERBOARD
2835						CMP	RO,MEMSIZ	;PATTERN
2836						BLOS	1\$	
2837						MOV	#625,CYLINDER	
2838	014562	012706	000500			CLR	DMA	
2839	014566	012767	015152	163230		MOV	#400,WRDCT	
2840	014574	012767	000340	163224		MOV	#OUTBUF,BUF	
2841	014602	004567	002030			TSTB	ARPCS	;WAIT FOR DONE
2842	014606	012777	000625	011034		BPL	.-4	
2843	014614	052777	000011	011016		TST	ARPDS	;WAIT FOR UNIT READY
2844	014622	012700	025762			BPL	.-4	
2845	014626	012720	025252			JSR	RS,FUNCT	;WRITE 1 SECTOR OF CHECKERBOARD
2846	014632	020067	005110		.WORD	3		
2847	014636	101773				TSTB	ARPCS	;WAIT FOR READY
2848	014640	012767	000625	011030		BPL	.-4	
2849	014646	005067	011026			BIT	#815,ARPCS	;ANY ERRORS?
2850	014652	012767	000400	011014		BEQ	2\$	
2851	014660	012767	025762	011020		HLT		;DEVICE ERROR ON WRITE
2852	014666	105777	010746		2\$:	MOV	#PFTST,LAD	
2853	014672	100375				SCOPE		
2854	014674	005777	010760			JSR	RS,PRINT\$	;PRINT MESSAGE
2855	014700	100375				MESS		;HAVE POWER TURNED OFF
2856	014702	004567	000274		3\$:	JSR	RS,FUNCT	;GO INTO A LOOP READING
2857	014706	000003			.WORD	5		;THE DISK SURFACE
2858	014710	105777	010724			TSTB	ARPCS	
2859	014714	100375				BPL	.-4	
2860	014716	032777	100000	010714		BR	3\$	
2861	014724	001401						
2862	014726	104400						
2863	014730	012767	014562	164044				
2864	014736	104000						
2865	014740	004567	164320					
2866	014744	023720						
2867	014746	004567	000230					
2868	014752	000005						
2869	014754	105777	010660					
2870	014760	100375						
2871	014762	000771						
2872								
2873								
2874								
2875								
2876	014764	012777	000001	010646	PFT1:	MOV	#1,ARPCS	;CLEAR THE CONTROLLER
2877	014772	016767	010666	010734		MOV	FLAG,WORK	;GET UNIT NUMBER
2878	015000	000241				CLC		
2879	015002	006067	010726			ROR	WORK	
2880	015006	006067	010722			ROR	WORK	
2881	015012	000367	010716			SWAB	WORK	
2882	015016	042767	174377	010710		BIC	#174377,WORK	

;AFTER MACHINE IS POWERED DOWN AND UP CONTROL
 ;IS TRANSFERRED HERE.

2883	015024	016777	010704	010606	MOV	WORK,ARPCS	;SELECT THE UNIT
2884	015032	005777	010622		TST	ARPCS	
2885	015036	100352			BPL	PFT1	
2886	015040	005777	010616		TST	ASUCA	;SELECTED CYLINDER SHOULD BE ZERO
2887	015044	001407			BEQ	1\$	
2888	015046	005067	002752		CLR	EXP\$	
2889	015052	017767	010604	002746	MOV	ASUCA,REC\$	
2890	015060	104401			HLT	+1	;HEADS DID NOT RESTORE TO 0
2891	015062	000427			BR	2\$	;ON POWER FAILURE
2892	015064	012700	025762	1\$:	MOV	#OUTBUF,RO	
2893	015070	022720	025252	4\$:	CMP	#25252,(RO)+	;DID MEMORY RETAIN PATTERN ON PF
2894	015074	001004			BNE	3\$	;BRANCH IF NO
2895	015076	020067	004644		CMP	RO,MEMSIZ	
2896	015102	001372			BNE	4\$	
2897	015104	000416			BR	2\$	
2898	015106	012767	025252	002710	MOV	#25252,EXP\$	
2899	015114	016067	177776	002704	MOV	-2(RO),REC\$	
2900	015122	104403			HLT	+3	;CONTENTS OF MEMORY CHANGED AFTER PF
2901	015124	004567	164134		JSR	R5,PRINT\$	;PRINT MESSAGE
2902	015130	023703			MES4		;PRINT MEMORY LOCATION
2903	015132	010067	164424		MOV	RO,TTY	
2904	015136	004767	164202		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
2905	015142			2\$:			
2906	015142	004567	164116		JSR	R5,PRINT\$	;PRINT MESSAGE
2907	015146	025625			END		;END OF TEST
2908	015150	000000			HALT		
2909							
2910							
2911							
2912	015152	012767	015162	162644	PFD: MOV	#PFU,24	;SET UP POWER UP VECTOR
2913	015160	000774			BR	PFD	
2914							
2915							
2916							
2917	015162	012767	000026	162634	PFU: MOV	#26,24	;SETUP TRAP CATCHER
2918	015170	005067	162632		CLR	26	
2919	015174	012706	000500		MOV	#STKPTR,SP	;RESTORE STACK
2920	015200	000671			BR	PFT1	;VERIFY POWER DOWN SEQUENCE
2921							

.SBTTL *** SUBROUTINES ***

 ;THIS ROUTINE OUTPUTS THE FUNCTION FOUND AT
 ;THE CALL + 2.

2922											
2923											
2924											
2925											
2926											
2927	015202	004567	001430			FUNCT:	JSR	RS,DSKNOS		;SELECT THE UNIT	
2928	015206	016777	010466	010436			MOV	DMA,ARPCA		;SETUP DISK ADDR REG	
2929	015214	016777	010456	010426			MOV	CYLINDER,ARPCA		;SETUP CYLINDER ADDR REG	
2930	015222	016777	010460	010416			MOV	BUF,ARPCA		;SETUP BUS ADDR REG	
2931	015230	016777	010440	010406			MOV	WRDCT,ARPCW		;SETUP WORD COUNT	
2932	015236	005477	010402				NEG	ARPCW		;COMPLIMENT WORD COUNT	
2933	015242	011567	000022				MOV	(RS),FNCT		;GET RPCS FUNCTION	
2934	015246	062705	000002				ADD	#2,RS		;UPDATE RETURN ADDR	
2935	015252	056767	004472	000010			BIS	MEX,FNCT		;ADD EXTENDED MEMORY BITS	
2936	015260	056777	000004	010352			BIS	FNCT,ARPCS		;OUTPUT THE FUNCTION	
2937	015266	000205					RTS	RS			
2938	015270	000000				FNCT:	0				

;RP11 DISK INTERRUPT HANDLER

2943	015272	005067	163742			DKINT:	CLR	ERRFLG		;CLEAR THE ERROR FLAG	
2944	015276	005777	010336				TST	ARPCS		;TEST FOR ERROR	
2945	015302	100402					BMI	1\$			
2946	015304	000167	000316				JMP	INTXT		;JUMP IF NO ERRORS	
2947	015310	010667	163724			1\$:	MOV	SP,ERRFLG		;SET INTERRUPT ERROR FLAG	
2948	015314	005767	010404				TST	RDERR		;IS THIS THE FIRST ERROR ATTEMPT?	
2949	015320	001404					BEQ	2\$		;BRANCH IF YES	
2950	015322	032737	000020	177570			BIT	#B4,ARPCW		;TYPE ALL ERROR ATTEMPTS?	
2951	015330	001522					BEQ	DKI1		;BRANCH IF NO	
2952	015332	104400				2\$:	HLT			;STATUS ERROR AFTER INTERRUPT	
2953	015334	032777	000002	010276			BIT	#B1,ARPCS		;CHECK FOR READ	
2954	015342	001012					BNE	DELMES		;BRANCH IF WRITTING	
2955	015344	004567	163714				JSR	RS,PRINTS		;PRINT MESSAGE	
2956	015350	024127					MES13			;GIVE # OF READ ATTEMPT	
2957	015352	016767	010346	164202			MOV	RDERR,TTY			
2958	015360	004767	163772				JSR	PC,PRINTS		;TYPE LOCATION-SUPRESS ZEROS	
2959	015364	010667	163650			3\$:	MOV	SP,ERRFLG			
2960											
2961	015370	017767	010254	000266		DELMES:	MOV	ARPCA,INT1		;GET CYLINDER ADDR	
2962	015376	017767	010250	000256			MOV	ARPCA,INT0		;GET HEAD AND SECTOR ADDR	
2963	015404	042767	160360	000250			BIC	#160360,INT0		;CLEAR UNWANTED BITS	
2964	015412	032777	000001	010236			BIT	#B0,ARPCW		;WAS IT AN ADDR ERROR?	
2965	015420	001026					BNE	REDAC		;BRANCH IF YES	
2966	015422	032767	000017	000232			BIT	#17,INT0		;IS SECTOR = TO 0	
2967	015430	001403					BEQ	DECTK		;YES - BRANCH	
2968	015432	005367	000224				DEC	INT0		;BACK UP COUNT	
2969	015436	000417					BR	REDAC			
2970	015440	132767	000037	000215		DECTK:	BITB	#37,INT0+1		;IS HEAD = TO 0	
2971	015446	001406					BEQ	DECCY		;YES - BRANCH	
2972	015450	105367	000207				DECB	INT0+1		;BACK UP HEAD	
2973	015454	052767	000011	000200			BIS	#11,INT0		;SET UP SECTOR	
2974	015462	000405					BR	REDAC			
2975	015464	012767	011411	000170		DECCY:	MOV	#11411,INT0			
2976	015472	005367	000166				DEC	INT1			
2977	015476					REDAC:					


```

2978 015476 004567 163562      JSR    RS,PRINT$      ;PRINT MESSAGE
2979 015502 024143      MES14      ;REPORT CYLINDER ADDR
2980 015504 016767 000154 164050  MOV    INT1,TTY
2981 015512 004767 163640      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
2982 015516 005067 000142      CLR     INT1
2983 015522 116767 000135 000134  MOV    INTO+1,INT1
2984 015530 004567 163530      JSR    RS,PRINT$      ;PRINT MESSAGE
2985 015534 024153      MES15      ;REPORT HEAD ADDR OF FAILURE
2986 015536 016767 000122 164016  MOV    INT1,TTY
2987 015544 004767 163606      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
2988 015550 116767 000106 000106  MOV    INTO,INT1
2989 015556 004567 163502      JSR    RS,PRINT$      ;PRINT MESSAGE
2990 015562 024164      MES16      ;REPORT SEC ADDR OF FAILURE
2991 015564 016767 000074 163770  MOV    INT1,TTY
2992 015572 004767 163560      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
2993 015576 032777 001000 010054  DK11:  BIT    #8,DRPDS      ;IS DRIVE UNSAFE?
2994 015604 001401      BEQ     .+4
2995 015606 000000      HALT
2996 015610 032777 002000 010042  BIT    #10,DRPDS      ;DRIVE UNSAFE
2997 015616 001403      BEQ     INTEXT      ;SEEK INCOMPLETE?
2998 015620 112777 000015 010012  MOV    #15,DRPCS      ;BRANCH IF COMPLETE
2999 015626 105777 010006      INTEXT: TST    DRPCS      ;RECALIBRATE
3000 015632 100375      BPL     .-4
3001 015634 005777 010020      TST    DRPDS      ;WAIT FOR DONE
3002 015640 100375      BPL     .-4
3003 015642 005767 004646      TST    BCKFLG      ;WAIT FOR READY
3004 015646 001402      BEQ     1$      ;DID WE COME FROM BACKGROUND TEST?
3005 015650 012716 022502      MOV    #NPRRET,(SP)      ;BRANCH IF NO
3006 015654 010667 166632      1$:  MOV    SP,INTFLG      ;MODIFY RETURN ADDR
3007 015660 000002      RTI      ;SET INTERRUPT OCCURRED FLAG
3008
3009 015662 000000      INTO:  0
3010 015664 000000      INT1:  0
3011
3012      ;THIS ROUTINE IF SWITCH 07 OFF ISSUES A SEEK RANDOM YET NOT DISTROYING
3013      ;THE CYLINDER UNDER TEST
3014
3015 015666 105737 177570      SEEK:  TST    @#SWR
3016 015672 100534      BMI     SEEND
3017 015674 004567 000736      JSR    RS,DSKNOS
3018 015700 005767 163334      TST    ERRFLG
3019 015704 001031      BNE     2$
3020 015706 062767 123455 010036  1$:  ADD    #123455,SEEK1      ;ISSUE A RANDOM SEEK?
3021 015714 042767 177000 010030      BIC    #177000,SEEK1      ;BR IF NO
3022 015722 022767 000625 010022      CMP    #625,SEEK1      ;SELECT THE UNIT
3023 015730 002766      BLT     1$      ;ARE WE LOOPING ON ERROR?
3024 015732 016745 010014      MOV    SEEK1,-(SP)      ;BR IF YES-DO SAME SEEK AS THE ERROR
3025 015736 166716 007734      SUB    CYLINDER,(SP)      ;GET A RANDOM NUMBER
3026 015742 100405      BMI     7$      ;CLEAR OUT UNWANTED CYLINDER BITS
3027 015744 021627 000200      CMP    (SP),#200      ;VALID CYLINDER NUMBER
3028      ;BR IF TOO BIG
3029 015750 100006      BPL     8$      ;PUT NEW CYLINDER ON STACK
3030 015752 005726      9$:  TST    (SP)+      ;CREAT A DIFFERENCE ON STACK
3031 015754 000754      BR      1$      ;BR IF DIFFERENCE IS -
3032 015756 021627 177600      7$:  CMP    (SP),#-200      ;IS DIFFERENCE BETWEEN CYLINDER AND RANDOM CYL
3033 015762 100401      BMI     8$      ;GREATER THAN 200 CYLINDERS
      ;YES, CLEANUP STACK AND CONTINUE
      ;NO, CLEAN OFF STACK
      ;GET NEW CYLINDER RANDOM NUMBER
      ;BR IF DIFFERENCE GRATER THAN 200 CLYINDERS

```


H05

```

3034 015764 000772      BR      9$      ;DIFFERENCE LESS THAN 200 GET ANOTHER CYL NO
3035 015766 005726      8$: TST      (SP)+      ;CLEANOFF STACK
3036 015770 017746 007640 2$: MOV      @VECTOR,-(SP)      ;SAVE VECTOR
3037 015774 017746 007636      MOV      @STATUS,-(SP)      ;SAVE PRIORITY
3038 016000 012777 016166 007626      MOV      @SEKDON,@VECTOR      ;SETUP TRAP FOR RANDOM SEEK
3039 016006 012777 000340 007622      MOV      #340,@STATUS
3040 016014 016777 007732 007626 3$: MOV      SEEK1,@RPCA      ;SET RANDOM CYLINDER
3041 016022 012777 177777 007630      MOV      #-1,@RPDS      ;CLEAR ATTENTIONS
3042 016030 042777 000100 007602      BIC      #86,@RPCS      ;CLEAR INTERRUPT DONE
3043 016036 052777 020011 007574      BIS      #20011,@RPCS      ;SEEK INTERRUPT ON ATTENTION
3044 016044 000001      WAIT
3045 016046 005777 007566      TST      @RPCS      ;ERROR?
3046 016052 100035      BPL      4$      ;BR IF NO ERROR
3047 016054 104400      HLT
3048 016056 032777 004000 007554      BIT      #811,@RPCS      ;DISK ERROR DURING RANDOM SEEK
3049 016064 001006      BNE      5$      ;SEEK INCOMPLETE?
3050 016066 032777 001000 007564      BIT      #89,@RPDS      ;BR IF SEEK INCOMPLETE
3051 016074 001402      BEG      5$      ;FILE UNSAFE?
3052 016076 000000      HALT      ;BR IF NOT UNSAFE
3053 016100 000410      BR      6$      ;FILE UNSAFE
3054 016102      5$:      6$      ;GO DO A HOME SEEK
3055 016102 004567 163156      JSR      R5,PRINT$      ;PRINT MESSAGE
3056 016106 024143      MES14      ;CYLINDER =
3057 016110 016767 007636 163444      MOV      SEEK1,TTY
3058 016116 004767 163234      JSR      PC,PRINT$      ;TYPE LOCATION-SUPPRESS ZEROS
3059 016122 012777 000015 007510 6$: MOV      #15,@RPCS      ;DO A HOME SEEK
3060 016130 105777 007504      TSTB      @RPCS
3061 016134 100375      BPL      -4      ;WAIT FOR DONE
3062 016136 005777 007516      TST      @RPDS
3063 016142 100375      BPL      -4      ;WAIT FOR READY
3064 016144 000723      BR      3$      ;TRY SEEK AGAIN
3065 016146 005777 007506      4$: TST      @RPDS
3066 016152 100375      BPL      -4      ;WAIT FOR READY
3067 016154 012677 007456      MOV      (SP)+,@STATUS      ;RESTORE TRAPS
3068 016160 012677 007450      MOV      (SP)+,@VECTOR
3069 016164 000207      SEEND: RTS      ;RETURN FROM RANDOM SEEK
3070 016166 000002      SEKDON: RTI      ;RANDOM SEEK DONE
3071
3072
3073      ;ROUTINE TO SET UP CYLINDER AND DISK ADDRESS FROM
3074      ;OPERATOR INPUTS DURING CONVERSATION MODE.
3075
3076 016170 032767 000040 007466 OPDSEL: BIT      #85,FLAG      ;USE OPERATOR ADDR?
3077 016176 001001      BNE      +4
3078 016200 000207      RTS      PC      ;NO
3079 016202 016767 007460 007466      MOV      SCYL,CYLINDER      ;GET CYLINDER ADDR
3080 016210 016767 007456 007462      MOV      SSEC,DMA      ;GET SECTOR ADDR
3081 016216 116767 007446 007455      MOVB      SHED,DMA+1      ;GET HEAD ADDR
3082 016224 000207      RTS      PC
3083
3084
3085
3086      ;ROUTINE TO SETUP DISK BUFFERS
3087      ;ADD WORD COUNT TO STARTING DISK ADDRESSES
3088      ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
3089

```



```

3090 016226 032767 000040 007430 DISBUF: BIT      #B5,FLAG      ;DID OPERATOR SUPPLY ADDR?
3091 016234 001401          BEQ      +4
3092 016236 000461          BR      BUFEXIT      ;OPERATOR DEFINED DISK ADDR
3093 016240 004767 000520          JSR      PC,BLSZ      ;DEFINE BLOCK SIZE
3094 016244 016767 007446 007464          MOV      BLOCK,WORK1
3095 016252 016767 007422 007454 INCSEC: MOV      DMA,WORK      ;GET DISK ADDR
3096 016260 042767 177760 007446          BIC      #177760,WORK      ;MASK OUT SECTOR COUNT
3097 016266 022767 000011 007440          CMP      #11,WORK      ;CHECK FOR LAST SECTOR
3098 016274 001406          BEQ      INCSUR      ;CHECK SURFACE
3099 016276 005267 007376          INC      DMA      ;+1 SECTOR COUNT
3100 016302 005367 007410          DECBLK: DEC      BLOCK      ;-1 FROM BLOCK COUNT
3101 016306 001432          BEQ      CMDAE      ;CMP DMA TO RPDA
3102 016310 000760          BR      INCSEC      ;RECYCLE
3103 016312 042767 000017 007360 INCSUR: BIC      #17,DMA      ;FETCH ADDRESS
3104 016320 016767 007354 007406          MOV      DMA,WORK
3105 016326 042767 160377 007400          BIC      #160377,WORK
3106 016334 122767 000023 007373          CMPB     #23,WORK+1
3107 016342 001403          BEQ      SWSUR      ;+1 SURFACE
3108 016344 105267 007331          INCB     DMA+1      ;INC HEAD NUMBER
3109 016350 000754          BR      DECBLK      ;RECYCLE
3110 016352 005067 007322          SWSUR: CLR      DMA      ;CLEAR THE DISK ADDRESS
3111 016356 005267 007314          INC      CYLINDER
3112 016362 022767 000626 007306          CMP      #626,CYLINDER      ;HAS LAST CYL BEEN EXCEEDED?
3113 016370 001404          BEQ      BUFEXIT      ;BRANCH IF YES
3114 016372 000743          BR      DECBLK
3115          ;COME HERE AFTER DETERMINING THE STARTING ADDR OF THE NEXT
3116          ;TRANSFER. NOW CHECK TO SEE THERE IS ENOUGH ROOM ON THE DISK
3117          ;TO MAKE THE TRANSFER. IF NOT MODIFY THE WORD COUNT FOR THE FINAL
3118          ;OUTPUT.
3119
3120
3121 016374 105767 007264          CMDAE: TSTB     FLAG      ;CHECK FOR LAST DISK BUFFER
3122 016400 100015          BPL      BUFINX
3123 016402 005067 007272          BUFEXIT: CLR      DMA      ;CLEAR ADDRESS BITS
3124 016406 005067 007264          CLR      CYLINDER      ;CLR CYLINDER REGISTER
3125 016412 062716 000002          ADD      #2,(6)      ;INC STACK POINTER
3126 016416 042767 000200 007240          BIC      #200,FLAG
3127 016424 016767 007260 007242          MOV      SWRDCT,WRDCT
3128 016432 000500          BR      EXTDR      ;EXIT
3129 016434 005067 007300          BUFINX: CLR      WORK2      ;CLEAR BLOCK COUNTER
3130 016440 016767 007234 007266          MOV      DMA,WORK
3131 016446 016767 007224 007266          MOV      CYLINDER,WORK3
3132 016454 042767 160360 007252          BIC      #160360,WORK
3133 016462 005267 007252          XINCSEC: INC     WORK2      ;INCREMENT BLOCK COUNT
3134 016466 005367 007244          DEC      WORK1      ;DECREMENT TOTAL BLOCKS REQUIRED
3135 016472 001460          BEQ      EXTDR      ;EXIT IF BLOCK COUNT SATISFIED
3136 016474 122767 000011 007232          CMPB     #11,WORK      ;CHECK THE DISK ADDRESS TO
3137 016502 001403          BEQ      XINCSUR      ;SEE IF THERE IS ENOUGH ROOM
3138 016504 005267 007224          INC      WORK      ;TO HANDLE THE OUTPUT REQUESTED
3139 016510 000764          BR      XINCSEC
3140 016512 105067 007216          XINCSUR: CLRB     WORK
3141 016516 122767 000023 007211          CMPB     #23,WORK+1
3142 016524 001403          BEQ      1$
3143 016526 105267 007203          INCB     WORK+1
3144 016532 000753          BR      XINCSEC
3145 016534 005067 007174          1$:      CLR      WORK

```



```

3146 016540 022767 000625 007174      CMP      #625,WORK3      ;ARE WE ON THE LAST CYLINDER?
3147 016546 001403                BEQ      2$              ;BRANCH IF YES
3148 016550 005267 007166                INC      WORK3
3149 016554 000742                BR       XINCSEC
3150 016556 016767 007156 007110 2$:    MOV      WORK2,WRDCT      ;COME HERE IF THERE IS NOT
3151 016564 000241                CLC                      ;ENOUGH ROOM TO HANDLE THE
3152 016566 006167 007102                ROL      WRDCT          ;REQUESTED OUTPUT. MODIFY THE
3153 016572 006167 007076                ROL      WRDCT          ;WORDCOUNT TO FILL THE REMAINING
3154 016576 006167 007072                ROL      WRDCT          ;SURFACE.
3155 016602 006167 007066                ROL      WRDCT
3156 016606 006167 007062                ROL      WRDCT
3157 016612 006167 007056                ROL      WRDCT
3158 016616 006167 007052                ROL      WRDCT
3159 016622 006167 007046                ROL      WRDCT
3160 016626 052767 000200 007030      BIS      #200,FLAG
3161 016634 000207                EXTDR:  RTS      PC              ;EXIT
3162
3163
3164                                ;ROUTINE TO SELECT THE DISK UNIT
3165
3166 016636 016767 007022 007070      DSKNOS: MOV      FLAG,WORK      ;FETCH THE FLAG WORD
3167 016644 006067 007064                ROR      WORK
3168 016650 006067 007060                ROR      WORK
3169 016654 000241                CLC
3170 016656 000367 007052                SWAB     WORK
3171 016662 042767 174377 007044      BIC      #174377,WORK      ;MASK THE DISK NUMBER
3172 016670 016777 007040 006742      MOV      WORK,ARPCS      ;LOAD THE ADDRESS IN THE ADDRESS REG
3173 016676 005777 006756                TST      ARPD$          ;IS THE UNIT READY?
3174 016702 100401                BMI      1$              ;BRANCH IF READY
3175 016704 104400                HLT
3176 016706 000205                1$:    RTS      R5              ;SELECTED UNIT NOT READY
3177                                ;EXIT
3178
3179                                ;INITIALIZE THE VECTORS
3180
3181 016710 012767 001112 161116      INIT:  MOV      #ERROR,34      ;SETUP TRAP VECTOR
3182 016716 012767 000340 161112      MOV      #PRI7,36
3183 016724 012767 001004 161076      MOV      #SCOPE$,30      ;SETUP EMT VECTOR
3184 016732 012767 000340 161072      MOV      #PRI7,32
3185 016740 012777 015272 006666      MOV      #DKINT,VECTOR      ;SETUP DISK INTERRUPT VECTOR
3186 016746 012777 000340 006662      MOV      #PRI7,STATUS
3187 016754 012737 000340 177776      MOV      #PRI7,#PSW      ;LOCKOUT INTERRUPTS
3188 016762 000207                RTS      PC
3189
3190                                ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
3191
3192
3193 016764 012767 000377 006724      BLSZ:  MOV      #377,BLOCK      ;DRIVE BLOCK SIZE
3194 016772 016767 006676 006734      MOV      WRDCT,WORK      ;FETCH WORD COUNT
3195 017000 036767 006712 006726      BIT      BLOCK,WORK
3196 017006 001410                BEQ      RORBLK
3197 017010 046767 006702 006716      BIC      BLOCK,WORK      ;SET UP BLOCK OVERFLOW
3198 017016 005267 006674                INC      BLOCK
3199 017022 066767 006670 006704      ADD      BLOCK,WORK
3200 017030 000367 006700                RORBLK: SWAB     WORK
3201 017034 016767 006674 006654      MOV      WORK,BLOCK      ;BLOCK COUNT

```



```

3202 017042 000207          RTS      PC          ;EXIT
3203
3204
3205          ; DETERMINE THE APPROPRIATE ATTENTION BIT FROM
3206          ; THE UNIT NUMBER.
3207
3208 017044 016701 006614    GATTN:  MOV     FLAG,R1
3209 017050 006001          ROR      R1
3210 017052 006001          ROR      R1          ;GET UNIT NUMBER
3211 017054 005067 000014    CLR      ATTN
3212 017060 042701 177770    BIC      #177770,R1      ;ISOLATE UNIT
3213 017064 116167 017076 000002  MOVB   ATTNB(R1),ATTN ;GET ATTENTION BIT
3214 017072 000207          RTS      PC
3215
3216
3217 017074 000000          ATTN:  0
3218 017076 001      002      004  ATTNB:  .BYTE  1,2,4,10,20,40,100,200
3219 017101 010      020      040
3220 017104 100      200
3221          .EVEN
3222
3223
3224
3225
3226          ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
3227
3228          ;ENTER FROM JSR PC PASEL
3229 017106 016700 006572    PASEL:  MOV     PATNU,RO      ;SET UP PATTERN NUMBER
3230 017112 016767 006556 006614    MOV     WRDCT,WORK    ;SET UP WORK
3231 017120 012701 025762    MOV     #OUTBUF,R1          ;LOC. OF OUT BUFFER
3232 017124 032767 000500 006532    BIT      #500,FLAG    ;BK WITH MEM. MANAG. OR NO MEM.MANAG.
3233 017132 001002          BNE      1$                  ;BR IF TRUE
3234 017134 012701 040000    MOV     #40000,R1          ;START OF MEM. MANAG. BUFFER
3235 017140 022700 000034    1$:    CMP     #34,RO        ;TEST FOR RANDOM DATA NUMBER
3236 017144 001423          BEQ      RANDOM              ;GO GENERATE RANDOM DATA
3237 017146 022700 000032    CMP     #32,RO        ;IS THIS PATTERN 15
3238 017152 001406          BEQ      PATT32
3239 017154 016021 020030    FILDAT: MOV     PAT0(0),(1)+  ;FILL BUFFER
3240 017160 005367 006550    DEC     WORK            ;DEC. WORK COUNT
3241 017164 001373          BNE     FILDAT              ;LOAD NEXT WORD
3242 017166 000207          RTS      PC                ;BUFFER FULL
3243 017170 012721 177777    PATT32: MOV     #177777,(1)+ ;INSERT ALL ONES PATTERN
3244 017174 005367 006534    DEC     WORK
3245 017200 001404          BEQ      1$
3246 017202 005021          CLR      (1)+              ;LOAD ZERO PATTERN
3247 017204 005367 006524    DEC     WORK            ;DECREMENT WORD COUNT
3248 017210 001367          BNE     PATT32              ;LOOP IF NOT ZERO
3249 017212 000207          1$:    RTS      PC          ;EXIT
3250          ;RANDOM DATA GENERATOR SUBROUTINE
3251 017214 016767 000134 000136  RANDOM: MOV     LONUN,LOSAV
3252 017222 016767 000130 000132    MOV     HINUN,HISAV
3253 017230 016700 000120    1$:    MOV     LONUN,RO      ;SET UP RO WITH 5 DIGITS LOW
3254 017234 016704 000116    MOV     HINUN,R4          ;SET UP R1 WITH 5 DIGITS HIGH
3255 017240 012703 000007    MOV     #7,R3            ;SET UP SHIFT COUNT
3256 017244 005002          CLR      R2                ;CLEAR R2
3257 017246 006300    2$:    ASL      RO                ;SHIFT RO LEFT AND

```


3258	017250	006104			ROL	R4	; ROTATE CARRY INTO LSB OF R1 INTO	
3259	017252	006102			ROL	R2	; ROTATE CARRY OUT OF R1 INTO R2	
3260	017254	005303			DEC	R3	; DECREMENT R3	
3261	017256	001373			BNE	2\$	; CONTINUE SHIFT LOOP	
3262	017260	066702	000070		ADD	LONUN,R2	; ADDN IN NUMBER TO MAKE X 129	
3263	017264	005504			ADC	R4	; PROPOGATE CARRY	
3264	017266	066704	000064		ADD	HINUN,R4	; ADDN IN NUMBER TO MAKE X 129	
3265	017272	005502			ADC	R2	; PROPOGATE CARRY	
3266	017274	062700	001057		ADD	#1057,R0	; ADDN LOW CONSTANT	
3267	017300	005504			ADC	R4	; PROPOGATE CARRIES	
3268	017302	005502			ADC	R2	; PROPOGATE AGAIN	
3269	017304	062704	047401		ADD	#47401,R4	; ADDN HIGH CONSTANT	
3270	017310	005502			ADC	R2	; PROPOGATE CARRY	
3271	017312	062702	000006		ADD	#6,R2	; ADDN HIGHEST CONSTANT	
3272	017316	060200			ADD	R2,R0	; REPRIME R0 WITH HIGH DIGIT	
3273	017320	005504			ADC	R4	; PROPOGATE CARRY	
3274	017322	010067	000026		MOV	R0,LONUN	; PUT R0 BACK IN LONUM	
3275	017326	010021			MOV	R0,(1)+	; HOLD LONUM FOR PROGRAM	
3276	017330	005367	006400		DEC	WORK		
3277	017334	001406			BEQ	EXGEN		
3278	017336	010467	000014		MOV	R4,HINUN	; PUT R1 BACK IN HINUM	
3279	017342	010421			MOV	R4,(1)+	; HOLD HINUM FOR PROGRAM	
3280	017344	005367	006364		DEC	WORK		
3281	017350	001327			BNE	1\$		
3282	017352	000207			RTS	PC	; RETURN TO PROGRAM	
3283	017354	000000			LONUN:	0		
3284	017356	000000			HINUN:	0		
3285	017360	000000			LOSAY:	0		
3286	017362	000000			HISAY:	0		
3287								
3288								
3289	017364	032767	000002	161650	MSG:	BIT	#B1,HLTCT\$	; TYPE ENTIRE MESSAGE
3290	017372	001103			BNE	1\$	; BRANCH IF NO	
3291	017374	004567	161664		JSR	R5,PRINT\$	; PRINT MESSAGE	
3292	017400	024030			MESB			
3293	017402	004567	161656		JSR	R5,PRINT\$	; PRINT MESSAGE	
3294	017406	023637			MES2A			
3295	017410	017767	006244	162144	MOV	QRPDS,TTY		
3296	017416	004767	161722		JSR	PC,PRINTR	; TYPE LOCATION WITH LEADING ZEROS	
3297	017422	004567	161636		JSR	R5,PRINT\$	; PRINT MESSAGE	
3298	017426	023615			MES1A			
3299	017430	017767	006222	162124	MOV	QRPDR,TTY		
3300	017436	004767	161702		JSR	PC,PRINTR	; TYPE LOCATION WITH LEADING ZEROS	
3301	017442	004567	161616		JSR	R5,PRINT\$	; PRINT MESSAGE	
3302	017446	023626			MES2			
3303	017450	017767	006164	162104	MOV	QRPCS,TTY		
3304	017456	004767	161662		JSR	PC,PRINTR	; TYPE LOCATION WITH LEADING ZEROS	
3305	017462	004567	161576		JSR	R5,PRINT\$	; PRINT MESSAGE	
3306	017466	023650			MES2B			
3307	017470	017767	006154	162064	MOV	QRPCA,TTY		
3308	017476	004767	161642		JSR	PC,PRINTR	; TYPE LOCATION WITH LEADING ZEROS	
3309	017502	004567	161556		JSR	R5,PRINT\$	; PRINT MESSAGE	
3310	017506	023661			MES2C			
3311	017510	017767	006136	162044	MOV	QRPDA,TTY		
3312	017516	004767	161622		JSR	PC,PRINTR	; TYPE LOCATION WITH LEADING ZEROS	
3313	017522	004567	161536		JSR	R5,PRINT\$	; PRINT MESSAGE	

[illegible]


```

3370          .EVEN
3371          ;RP11 DATA PATTERNS
3372
3373 020030 163126 PAT0: 163126
3374 020032 052525 PAT1: 052525
3375 020034 125252 PAT2: 125252
3376 020036 031463 PAT3: 031463
3377 020040 007417 PAT4: 007417
3378 020042 010421 PAT5: 010421
3379 020044 021042 PAT6: 021042
3380 020046 042104 PAT7: 042104
3381 020050 104210 PAT10: 104210
3382 020052 167356 PAT11: 167356
3383 020054 156735 PAT12: 156735
3384 020056 135673 PAT13: 135673
3385 020060 073567 PAT14: 073567
3386 020062 000001 PAT15: 000001
3387          ;PAT16 RANDOM DATA
3388
3389          ;THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
3390          ;ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 5 OF THE SWITCH
3391          ;REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN
3392          ;ERROR HAS BEEN FOUND AND WILL REPORT UP TO 3 VERIFY ERRORS
3393          ;WITHIN THE SAME INPUT OPERATION.
3394 020064 012767 177775 005640 COMPAR: MOV    #-3,ERCOUNT    ;ERROR RETRY COUNTER
3395 020072 016767 005576 005640      MOV    WRDCT,WORK2    ;GET THE WORD COUNT
3396 020100 012767 025762 005604      MOV    #OUTBUF,SAVE    ;SET UP OUTBUFFER POINTER
3397 020106 032767 000500 005550      BIT     #500,FLAG      ;BK WITH MEM. MANAG. OR NO MEM. MANAG
3398 020114 001003          BNE     1$          ;BR IF TRUE
3399 020116 012767 040000 005566      MOV    #40000,SAVE    ;START OF MEM. MANAG. OUT BUFFER
3400 020124 005067 005552 1$:      CLR     SWITCH      ;CLEAR RANDOM PATTERN FLAG
3401 020130 016767 177224 161544      MOV    LOSAV,LONUM    ;GET RANDOM BASE NOS.
3402 020136 016767 177220 161534      MOV    HISAV,HINUM
3403 020144 022767 000034 005532      CMP     #34,PATNU    ;IS THIS RANDOM PATTERN?
3404 020152 001422          BEQ     CMPLP      ;BRANCH IF YES
3405 020154 022767 000032 005522      CMP     #32,PATNU    ;IS THIS SPECIAL PATTERN?
3406 020162 001037          BNE     CMPLP1     ;BRANCH IF NO
3407 020164 005767 005512          CMPLP2: TST    SWITCH
3408 020170 001006          BNE     1$
3409 020172 012767 177777 177624      MOV    #177777,EXP$    ;EXPECT ALL ONES
3410 020200 010667 005476          MOV    SP,SWITCH    ;SET THE FLAG
3411 020204 000433          BR      WRDCMP      ;GO COMPARE DATA
3412 020206 005067 005470 1$:      CLR     SWITCH
3413 020212 005067 177606          CLR     EXP$      ;EXPECT ALL ZEROS
3414 020216 000426          BR      WRDCMP      ;GO COMPARE DATA
3415 020220 005767 005456          CMPLP: TST    SWITCH
3416 020224 001010          BNE     2$
3417 020226 004767 161332          JSR     PC,RAND$    ;GENERATE TWO RANDOM NOS.
3418 020232 016767 161444 177564      MOV    LONUM,EXP$    ;GET EVEN RANDOM WORD
3419 020240 010667 005436          MOV    SP,SWITCH    ;SET RANDOM PATTERN FLAG
3420 020244 000413          BR      WRDCMP
3421 020246 005067 005430 2$:      CLR     SWITCH
3422 020252 016767 161422 177544      MOV    HINUM,EXP$
3423 020260 000405          BR      WRDCMP
3424 020262 016700 005416          CMPLP1: MOV    PATNU,RO
3425 020266 016067 020030 177530      MOV    PAT0(RO),EXP$

```


3426	020274	027767	005412	177522	WRDCMP:	CMP	SAVE,EXP5	;COMPARE DATA
3427	020302	001021				BNE	WDERR	;WORD IN ERROR
3428	020304	005367	005430		WRDINC:	DEC	WORK2	;DECREMENT THE WORD COUNT
3429	020310	001415				BEQ	ADAM	;EXIT ROUTINE IF ZERO
3430	020312	062767	000002	005372	BLAD1:	ADD	#2,SAVE	;UPDATE PATTERN ADDRESS
3431	020320	022767	000032	005356		CMP	#32,PATNU	
3432	020326	101362				BHI	WRDCMP	;BRANCH IF STANDARD PATTERN
3433	020330	022767	000034	005346		CMP	#34,PATNU	;IS THIS RANDOM PATTERN
3434	020336	001730				BEQ	CMPLP	;BRANCH IF YES
3435	020340	000711				BR	CMPLP2	;BRANCH IF YES
3436	020342	000754				BR	WRDCMP	;COMPARE NEXT WORD
3437	020344	000207			ADAM:	RTS	PC	;EXIT THIS ROUTINE
3438	020346	005767	164140		WDERR:	TST	INTFLG	;DID INTERRUPT OCCUR YET?
3439	020352	001750				BEQ	WRDCMP	;BRANCH IF NO
3440	020354	017767	005332	177444		MOV	SAVE,RECS	;GET GOOD DATA
3441	020362	010667	160652			MOV	SP,ERRFLG	;SET ERROR FLAG
3442	020366	005767	005332			TST	RDERR	;IS THIS THE FIRST READ ERROR?
3443	020372	001404				BEQ	3\$	;BRANCH IF YES
3444	020374	032737	000020	177570		BIT	#B4,#SWR	;PRINT ALL RETRY ERRORS?
3445	020402	001556				BEQ	1\$	;BRANCH IF NO
3446	020404	032767	000100	005252	3\$:	BIT	#B6,FLAG	;MEMORY MANAGEMENT?
3447	020412	001002				BNE	10\$	;BR IF NO
3448	020414	104405				HLT	+5	;DATA COMPARE ERROR USING MEM. MANAG.
3449	020416	000401				BR	9\$	;GET AROUND HLT+3
3450	020420	104403			10\$:	HLT	+3	;DATA COMPARE ERROR
3451	020422	005067	005270		9\$:	CLR	BLOCK	;CLEAR THE BLOCK COUNTER
3452	020426	016767	005242	005300		MOV	WRDCT,WORK	;GET THE WORD COUNT
3453	020434	166767	005300	005272		SUB	WORK2,WORK	;DETERMINE DISTANCE OF FAILURE INTO BUFFER
3454	020442	162767	000400	005264	2\$:	SUB	#400,WORK	
3455	020450	100403				BMI	8\$	
3456	020452	005267	005240			INC	BLOCK	;UPDATE BLOCK COUNT FOR EACH 400 WORDS
3457	020456	000771				BR	2\$	
3458	020460	062767	000400	005246	8\$:	ADD	#400,WORK	;RESTORE POSITIVE NUMBER
3459	020466	016767	005206	005242		MOV	DMA,WORK1	;GET HEAD AND SECTOR ADDRESS
3460	020474	016767	005176	005240		MOV	CYLINDER,WORK3	;GET CYLINDER ADDRESS
3461	020502	005767	005210		5\$:	TST	BLOCK	;IS THE BLOCK COUNT ZERO?
3462	020506	001427				BEQ	7\$	;BRANCH IF YES
3463	020510	005367	005202			DEC	BLOCK	;DECREMENT BLOCK COUNT
3464	020514	122767	000011	005214		CMPB	#11,WORK1	;DETERMINE THE CYLINDER, HEAD,
3465	020522	001403				BEQ	4\$	;AND SECTOR ADDRESSES OF THE
3466	020524	005267	005206			INC	WORK1	;COMPARE ERROR
3467	020530	000764				BR	5\$	
3468	020532	105067	005200		4\$:	CLRB	WORK1	
3469	020536	122767	000023	005173		CMPB	#23,WORK1+1	
3470	020544	001403				BEQ	6\$	
3471	020546	105267	005165			INCB	WORK1+1	
3472	020552	000753				BR	5\$	
3473	020554	005067	005156		6\$:	CLR	WORK1	
3474	020560	005267	005156			INC	WORK3	
3475	020564	000746				BR	5\$	
3476	020566				7\$:			
3477	020566	004567	160472			JSR	RS,PRINTS	;PRINT MESSAGE
3478	020572	024143				MES14		;GIVE CYL ADDR
3479	020574	016767	005142	160760		MOV	WORK3,TTY	
3480	020602	004767	160550			JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
3481	020606	005067	005114			CLR	ACNVX	

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 66
 DZRPBE.P11 29-APR-77 14:41 *** SUBROUTINES ***

3482	020612	116767	005121	005106	MOV	WORK1+1,ACNVX	
3483	020620	004567	160440		JSR	RS,PRINTS	;PRINT MESSAGE
3484	020624	024153			MES15		;GIVE HEAD ADDR
3485	020626	016767	005074	160726	MOV	ACNVX,TTY	
3486	020634	004767	160516		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
3487	020640	116767	005072	005060	MOV	WORK1,ACNVX	
3488	020646	004567	160412		JSR	RS,PRINTS	;PRINT MESSAGE
3489	020652	024164			MES16		;GIVE SECTOR ADDR
3490	020654	016767	005046	160700	MOV	ACNVX,TTY	
3491	020662	004767	160470		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
3492	020666	004567	160372		JSR	RS,PRINTS	;PRINT MESSAGE
3493	020672	024043			MES9		
3494	020674	016767	005034	005024	MOV	WORK,ACNVX	;GET WORD COUNT INTO SECTOR
3495	020702	005267	005020		INC	ACNVX	
3496	020706	016767	005014	160646	MOV	ACNVX,TTY	
3497	020714	004767	160436		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
3498	020720	004567	160340		JSR	RS,PRINTS	;PRINT MESSAGE
3499	020724	024127			MES13		
3500	020726	016767	004772	160626	MOV	RDERR,TTY	
3501	020734	004767	160416		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
3502	020740	032737	000040	177570	1\$: BIT	#B5,#SWR	;CONTINUE COMPARING?
3503	020746	001405			BEQ	11\$	;BRANCH IF NO
3504	020750	005267	004756		INC	ERCOUNT	;UPDATE ERROR COUNTER
3505	020754	001402			BEQ	11\$	
3506	020756	000167	177322		JMP	WRDINC	
3507	020762	000167	177356		11\$: JMP	ADAM	
3508							
3509							
3510							
3511							
3512							
3513							
3514							
3515							
3516							
3517	020766	032767	000100	004670	EXTMEN: BIT	#B6,FLAG	;MEMORY MANAGEMENT IN USE?
3518	020774	001002			BNE	1\$	;BR IF NO
3519	020776	005037	177572		CLR	#SR0	;TURN IT OFF
3520	021002	052777	000001	004630	1\$: BIS	#B0,#RPCS	;CLEAR THE DISK
3521	021010	105777	004624		TSTB	#RPCS	
3522	021014	100375			BPL	.-4	
3523	021016	012737	000340	177776	MOV	#PRI7,#PSW	;LOCK UP PRIORITY LEVELS
3524	021024	012767	021074	156752	MOV	#MAXREF,4	;SET UP I/O BUS TRAP
3525	021032	012767	000340	156746	MOV	#PRI7,6	
3526	021040	012767	017446	004644	MOV	#17446,SAVE	;SET UP FOR 4K
3527	021046	005777	004640		EXREF: TST	#SAVE	;REFERENCE MEMORY
3528	021052	022767	157446	004632	CMP	#157446,SAVE	;TEST FOR 28K
3529	021060	001001			BNE	1\$	;BRANCH IF LESS THAN 28K
3530	021062	000407			BR	MAXRF1	;LAST REFERENCE MADE TO I/O REG.
3531	021064	062767	020000	004620	1\$: ADD	#20000,SAVE	;SET UP FOR NEXT MEMORY REF.
3532	021072	000765			BR	EXREF	;GO REFERENCE MEMORY
3533							
3534							
3535							
3536	021074	162767	020000	004610	MAXREF: SUB	#20000,SAVE	
3537	021102	012767	000006	156674	MAXRF1: MOV	#6,4	;RESTORE I/O BUS TRAP

;EXTENDED MEMORY EXERCISER
 ;THE PROGRAM DETERMINES HOW MUCH MEMORY
 ;IS ON THE SYSTEM THEN IT
 ;GENERATES A RANDOM BUFFER THAT SIZE
 ;AND WRITES AND WRITE CHECKS THE DATA

;ENTER HERE WHEN I/O BUS ERROR OCCURS


```

3538 021110 005067 156672 CLR 6
3539 021114 005737 000042 TST 2#42 ;UNDER MONITOR CONTROL?
3540 021120 001403 BEQ 1$ ;BRANCH IF NO
3541 021122 162767 005670 004562 SUB #3000,SAVE ;ALLOW ROOM FOR THE MONITOR
3542 021130 016767 004556 000610 1$: MOV SAVE,MMSIZ ;SAVE THE MAXIMUM MEMORY ADDRESS
3543 021136 000434 BR MMSIZE ;GO SIZE WITH MEMORY MANAGEMENT
3544 021140 162767 025762 004544 MM8K: SUB #OUTBUF,SAVE ;DETERMINE THE BUFFER SIZE
3545 021146 000241 CLC
3546 021150 006067 004536 ROR SAVE ;FORM WORD COUNT
3547 021154 016767 004532 004512 MOV SAVE,WRDCT ;SAVE IT
3548 021162 042767 000001 004522 BIC #80,SAVE ;MAKE ADDRESS EVEN
3549 021170 012767 025762 004524 MOV #OUTBUF,INBUF ;START OF INPUT BUFFER
3550 021176 066767 004510 004516 ADD SAVE,INBUF
3551 021204 000241 CLC
3552 021206 042767 000377 004460 BIC #377,WRDCT ;DETERMINE MAXIMUM WORD COUNT
3553 021214 016767 004454 004466 MOV WRDCT,SWRDCT
3554 021222 012706 000476 EXIT: MOV #STKPTR-2,SP
3555 021226 000205 RTS R5
3556
3557 ;THIS SUB-ROUTINE MAPS FOR MEMORY MANAGEMENT AND SETS UP
3558 ;BUFFER SIZES WITH MEMORY MANAGEMENT
3559
3560 021230 032767 000100 004426 MMSIZE: BIT #B6,FLAG ;USING MEMORY MANAGEMENT?
3561 021236 001340 BNE MM8K ;BR IF NO
3562 021240 005237 177572 INC 2#SRO ;TURN MEMORY MANAGEMENT BACK ON
3563 021244 012737 021312 000004 MOV #NXM,2#ERRVEC ;SET UP FOR TRAP
3564 021252 012737 000340 000006 MOV #340,2#ERRVEC+2
3565 021260 012737 000400 172344 MOV #400,2#KIPAR2 ;SETUP FOR NEXT 4K PAGE
3566 021266 012737 177406 172304 MOV #400*256.-400+UP,RW,2#KIPDR2 ;SET KIPDR2=RW UP 400 BLOCKS
3567 021274 012700 040000 MOV #40000,R0 ;SET UP FOR TEST
3568 021300 005710 1$: TST (R0)
3569 021302 062737 000040 172344 ADD #40,2#KIPAR2 ;JUMP PAGE BY 1K
3570 021310 000773 BR 1$ ;TEST NEXT PAGE
3571 021312 022737 000400 172344 NXM: CMP #400,2#KIPAR2 ;8K MACHINE?
3572 021320 001013 BNE 1$ ;BR IF MORE THAN 8K
3573 021322 052767 000400 004334 BIS #B8,FLAG ;8K MACHINE WITH MEMORY MANAGEMENT
3574 021330 005037 172304 CLR 2#KIPDR2 ;SET KIPDR2 TO BE NON EXISTANT
3575 021334 012737 000006 000004 MOV #6,2#4 ;RESTORE TRAP CATCHER
3576 021342 005037 000006 CLR 2#6
3577 021346 000674 BR MM8K ;FINISH MAPPING AS IF NO MEMORY MANAGEMENT
3578 021350 013767 172344 000400 1$: MOV 2#KIPAR2,PARMAX ;PUT KIPAR2 INTO WORK LOCATION
3579 021356 162767 000040 000372 SUB #40,PARMAX ;ADJUST MAXIMUM RELOCATION CONSTANT
3580 021364 016700 000366 MOV PARMAX,R0 ;GET WORKING COPY OF THE RELOCATION
3581 ;CONSTANT AND
3582 021370 042700 177637 BIC #177637,R0 ;FORM AN
3583 021374 006200 ASR R0 ;INDEX INTO
3584 021376 006200 ASR R0 ;THE TABLES FOR
3585 021400 006200 ASR R0 ;RELOCATION CONSTANT
3586 021402 006200 ASR R0 ;AND FOR THE SIZE OF THE LAST PAGE
3587 021404 166067 021760 000344 SUB PARTAB(R0),PARMAX ;CALCULATE THE MAXIMUM RELOCATION
3588 ;CONSTANT FOR MEMORY MANAGEMENT
3589 021412 016067 021770 000332 MOV MMTAB(R0),MMSIZ ;GET THE SIZE OF THE LAST PAGE AND
3590 ;GO CREATE BUFFER SIZES AND WORD COUNTS
3591 021420 022767 001600 000330 MMFIN: CMP #1600,PARMAX ;PARMAX >OR= 28K
3592 021426 003441 BLE 5$ ;BR IF YES, ELSE PROCEED ONWARD
3593 021430 005001 CLR R1 ;RESET FULL 4K BUFFER COUNTER

```


3594	021432	005067	004236		CLR	WRDCT	: CLEAR WORD COUNT
3595	021436	012700	000400		MOV	#400, R0	: SET UP TO EXTRACT THE NUMBER OF
3596							: FULL 4K BUFFERS FOR A PARTIAL WRDCT
3597	021442	020067	000310	1\$:	CMP	R0, PARMAX	: DO WE HAVE THE NUMBER OF FULL 4K PAGES?
3598	021446	001404			BEQ	2\$	: BR IF YES, ELSE REPEAT
3599	021450	062700	000200		ADD	#200, R0	: UPDATE TO NEXT 4K BUFFER
3600	021454	005201			INC	R1	: UPDATE FULL 4K BUFFER COUNTER
3601	021456	000771			BR	1\$	: AND REPEAT
3602	021460	010167	000270	2\$:	MOV	R1, PDRS	: STORE NUMBER OF FULL 4K BUFFERS
3603	021464	005701		3\$:	TST	R1	: FINISHED CALCULATING PARTIAL WORD COUNT?
3604	021466	001405			BEQ	4\$	: BR IF YES, ELSE PROCEED ON
3605	021470	062767	010000	004176	ADD	#10000, WRDCT	: USING THE NUMBER OF FULL 4K PAGES
3606	021476	005301			DEC	R1	: CALCULATE A PARTIAL WORD COUNT
3607	021500	000771			BR	3\$	
3608	021502	005267	000246	4\$:	INC	PDRS	: ADJUST FOR MEMORY MANAGEMENT MAPPING
3609	021506	066767	000240	004160	ADD	MMSIZ, WRDCT	: ADD SIZE OF LAST PAGE
3610	021514	016767	004154	004166	MOV	WRDCT, SWRDC	: AND SAVE TOTAL BUFFER SIZE AND
3611	021522	016767	004146	000250	MOV	WRDCT, MAXWC	: SAVE AGAIN FOR CONFORMITY
3612	021530	000414			BR	SALDR	: GO SAVE THE LOADERS
3613	021532	012767	050000	000240	5\$:	MOV	#50000, MAXWC
3614	021540	012767	040000	004142	MOV	#40000, SWRDC	: 20K TRANSFERS
3615	021546	066767	000200	004134	ADD	MMSIZ, SWRDC	: SET UP HIGHEST TRANSFER SIZE
3616	021554	012767	000005	000172	MOV	#5, PDRS	: ADD LAST PAGE SIZE
3617	021562	005767	004170		MOV	LDRFLG	: SET UP COUNT FOR KIPDRS
3618	021566	001017			TST	LDRFLG	: ARE THE LOADERS ALREADY RELOCATED?
3619	021570	005037	177572		1\$:	BNE	1\$
3620	021574	005167	004156		CLR	#SRO	: BR IF YES
3621	021600	016700	000142		COM	LDRFLG	: TURN OFF MEMORY MANAGEMENT
3622	021604	042700	003777		MOV	MEMSIZ, R0	: ADJUST LOADER RELOCATED FLAG
3623	021610	012737	021626	000004	BIC	#3777, R0	
3624	021616	012701	034000		MOV	#1\$ #ERRVEC	: SET UP FOR 1K WORD
3625	021622	012021			MOV	#34000, R1	: SETUP TRAP CATCHER
3626	021624	000776			2\$:	MOV	(R0)+, (R1)+
3627	021626	012737	000006	000004	BR	2\$	: 8K START ADDRESS FOR LOADERS
3628	021634	005037	000006		1\$:	MOV	#6, #ERRVEC
3629	021640	052737	000001	177572	CLR	#ERRVEC+2	: MOVE LOADER
3630	021646	016700	000102		BIS	#80, #SRO	: RESTORE TRAPCATCHER
3631	021652	005300			MOV	PDRS, R0	: TURN MEMORY MANAGEMENT BACK ON IF OFF
3632	021654	001422			DEC	R0	: PUT NUMBER OF KIPDRS INTO WORK REGISTER
3633	021656	012737	177406	172306	BEQ	3\$	: SETUP FOR PROPER NUMBER OF KIPDRS
3634	021664	005300			MOV	#400*256.-400+UP, RW, #KIPDR3	: BR IF NO KIPDRS NEED SETTING UP
3635	021666	001415			DEC	R0	: SET KIPDR3=RW UP 400 BLOCKS
3636	021670	012737	177406	172310	BEQ	3\$	: ENOUGH KIPDRS?
3637	021676	005300			MOV	#400*256.-400+UP, RW, #KIPDR4	: BR IF ENOUGH
3638	021700	001410			DEC	R0	: SET KIPDR4=RW UP 400 BLOCKS
3639	021702	012737	177406	172312	BEQ	3\$	: ENOUGH KIPDRS?
3640	021710	005300			MOV	#400*256.-400+UP, RW, #KIPDR5	: BR IF YES
3641	021712	001403			DEC	R0	: SET KIPDR5=RW UP 400 BLOCKS
3642	021714	012737	177406	172314	BEQ	3\$	: ENOUGH KIPDRS?
3643	021722				MOV	#400*256.-400+UP, RW, #KIPDR6	: BR IF YES
3644	021722	004567	157354		3\$:	JSR	RS, PRNTRF\$
3645	021726	024301			MES22		: SET KIPDR6=RW UP 400 BLOCKS
3646							: FORCE PRINT THE MESSAGE
3647	021730	012767	023514	157624	MOV	#RELOAD, TTY	: LOADERS HAVE BEEN RELOCATED TO 8K
3648	021736	004767	157436		JSR	PC, PRINTB	: TO RESTORE LOADERS START AT LOCATION
3649	021742	000167	177254		JMP	EXIT	: FORCE TYPE LOCATION - SUPPRESS ZEROS
							: MEMORY MANAGEMENT DONE


```

3650
3651 021746 000000 MEMSIZ: 0
3652 021750 000000 MEX: 0 ;CONTROLLER EXTENDED MEMORY ADDRESS
3653 021752 000000 MMSIZ: 0 ;THE LAST PAGE SIZE
3654 021754 000000 PDRS: 0 ;THE NUMBER OF PAGES IN THE BUFFER
3655 021756 000000 PARMAX: 0 ;THE MAXIMUM KIPAR VALUE
3656
3657 ;TABLE OF OFFSETS FOR CORRECTING PARMAX RELOCATION CONSTANT
3658 021760 000000 PARTAB: .WORD 0
3659 021762 000040 .WORD 40
3660 021764 000100 .WORD 100
3661 021766 000140 .WORD 140
3662
3663 ;TABLE OF LAST PAGE SIZES
3664 021770 002000 MMTAB: .WORD 1024.
3665 021772 004000 .WORD 2048.
3666 021774 006000 .WORD 3072.
3667 021776 010000 .WORD 4096.
3668
3669 022000 000000 MAXWC: 0
3670
3671 ;SUB-ROUTINE TO INITIALIZE KIPAR2-6
3672
3673 022002 016700 177746 PARINT: MOV PDRS,RO ;GET NUMBER OF KIPARS
3674 022006 012737 000400 172344 MOV #400,2#KIPAR2 ;SETUP FIRST PAGE
3675 022014 005300 DEC RO ;DONE?
3676 022016 001422 BEQ 1$ ;BR IF YES
3677 022020 012737 000600 172346 MOV #600,2#KIPAR3 ;SET UP SECOND PAGE
3678 022026 005300 DEC RO ;DONE?
3679 022030 001415 BEQ 1$ ;BR IF YES
3680 022032 012737 001000 172350 MOV #1000,2#KIPAR4 ;SETUP THIRD PAGE
3681 022040 005300 DEC RO ;DONE?
3682 022042 001410 BEQ 1$ ;BR IF YES
3683 022044 012737 001200 172352 MOV #1200,2#KIPAR5 ;SETUP FOURTH PAGE
3684 022052 005300 DEC RO ;DONE?
3685 022054 001403 BEQ 1$ ;BR IF YES
3686 022056 012737 001400 172354 MOV #1400,2#KIPAR6 ;SETUP FIFTH PAGE
3687 022064 062716 000004 1$: ADD #4,(SP)
3688 022070 000207 RTS PC
3689
3690 ;SUB-ROUTINE TO INCREMENT EACH KIPAR USED BY 1K
3691
3692 022072 005000 PARINC: CLR RO ;SETUP INDEX
3693 022074 016746 177654 MOV PDRS,-(SP) ;GET NUMBER OF KIPARS
3694 022100 006316 ASL (SP) ;DOUBLE IT
3695 022102 062760 000040 172344 1$: ADD #40,KIPAR2(RO) ;INCREMENT KIPAR BY 1K
3696 022110 005720 TST (RO)+ ;BUMP INDEX
3697 022112 020016 CMP RO,(SP) ;DONE?
3698 022114 001372 BNE 1$ ;BR IF NO
3699 022116 005726 TST (SP)+ ;CLEAN UP STACK
3700 022120 000207 RTS PC ;RETURN
3701
3702 ;SUB-ROUTINE TO GENERATE "MEX", "WRDCT", "BUF" USING MEMORY MANAGEMENT
3703
3704 022122 013700 172344 PARREG: MOV 2#KIPAR2,RO ;GET THE LOW REGISTER
3705 022126 042700 171770 BIC #171770,RO ;CLEAR ALL BUT ADDRESS BITS 17:19

```


G06

3706	022132	006300		ASL	RO		; PROPERLY POSITION MEX BITS
3707	022134	006300		ASL	RO		
3708	022136	000300		SWAB	RO		
3709	022140	010067	177604	MOV	RO, MEX		; STORE FOR DRIVE EXTENDED MEMORY BITS
3710	022144	013700	172344	MOV	2#KIPAR2, RO		; GET THE LOW REGISTER AGAIN
3711	022150	042700	006000	BIC	#6000, RO		; CLEAR ADDRESS BITS 17&18
3712	022154	006200		ASR	RO		; PROPERLY POSITION AS RPBA
3713	022156	006200		ASR	RO		
3714	022160	000300		SWAB	RO		
3715	022162	010067	003520	MOV	RO, BUF		; STORE AS BUS ADDRESS
3716	022166	042767	001000 003470	BIC	#B9, FLAG		; CLEAR LAST PAGE FLAG
3717	022174	005000		CLR	RO		; INITIALIZE INDEX
3718	022176	016700	177552	MOV	PDRS, RO		; GET NUMBER OF KIPARS
3719	022202	005300		DEC	RO		
3720	022204	006300		ASL	RO		; FINAL INDEX VALUE
3721	022206	016046	172344	MOV	KIPAR2(RO), -(SP)		; PUT VALUE OF LAST KIPAR ON STACK
3722	022212	022667	177540	CMP	(SP)+, PARMAX		; WORKING WITH LAST PAGE?
3723	022216	001003		BNE	1\$		; BR IF NOT LAST PAGE
3724	022220	052767	001000 003436	BIS	#B9, FLAG		; SET LAST PAGE BIT
3725	022226	032767	000002 003430 1\$:	BIT	#B1, FLAG		; OPERATOR SELECTING TRANSFER SIZE
3726	022234	001010		BNE	2\$		; BR IF OPERATOR SELECTING TRANSFER SIZE
3727	022236	032767	001000 003420	BIT	#B9, FLAG		; LAST TRANSFER?
3728	022244	001004		BNE	2\$		; BR IF YES
3729	022246	016767	177526 003420	MOV	MAXWC, WRDCT		; SETUP WORD COUNT
3730	022254	000403		BR	3\$		; EXIT FROM SUB-ROUTINE
3731	022256	016767	003426 003410 2\$:	MOV	SWRDCT, WRDCT		; SETUP OPERATOR OR LAST PAGE WORD COUNT
3732	022264	000207	3\$:	RTS	PC		; RETURN

; BACKGROUND TEST FOR INTERRUPTS - WORST CASE NPRS AND BUS PATTERN

3738	022266	010667	000222	NPR:	MOV	SP, BCKFLG	; SET BACKGROUND FLAG
3739	022272	012767	030000 000210		MOV	#30000, NPRCNT	; SETUP TIMEOUT COUNTER
3740	022300	012701	022512		MOV	#NPR1, R1	
3741	022304	005011			CLR	(R1)	; SETUP BUS PATTERN
3742	022306	000261			SEC		
3743	022310	000010		2\$:			
3744	022310	106111			ROLB	(R1)	
3745	022312	105421			NEGB	(R1)+	
3746	022314	105441			NEGB	-(R1)	
3747	022316	106111			ROLB	(R1)	
3748	022320	105421			NEGB	(R1)+	
3749	022322	105441			NEGB	-(R1)	
3750	022324	106111			ROLB	(R1)	
3751	022326	105421			NEGB	(R1)+	
3752	022330	105441			NEGB	-(R1)	
3753	022332	106111			ROLB	(R1)	
3754	022334	105421			NEGB	(R1)+	
3755	022336	105441			NEGB	-(R1)	
3756	022340	106111			ROLB	(R1)	
3757	022342	105421			NEGB	(R1)+	
3758	022344	105441			NEGB	-(R1)	
3759	022346	106111			ROLB	(R1)	
3760	022350	105421			NEGB	(R1)+	
3761	022352	105441			NEGB	-(R1)	

3762	022354	106111		ROLB	(R1)	
3763	022356	105421		NEGB	(R1)+	
3764	022360	105441		NEGB	-(R1)	
3765	022362	106111		ROLB	(R1)	
3766	022364	105421		NEGB	(R1)+	
3767	022366	105441		NEGB	-(R1)	
3768	022370	005201		INC	R1	
3769	022372	106111		ROLB	(R1)	
3770	022374	105421		NEGB	(R1)+	
3771	022376	105441		NEGB	-(R1)	
3772	022400	106111		ROLB	(R1)	
3773	022402	105421		NEGB	(R1)+	
3774	022404	105441		NEGB	-(R1)	
3775	022406	106111		ROLB	(R1)	
3776	022410	105421		NEGB	(R1)+	
3777	022412	105441		NEGB	-(R1)	
3778	022414	106111		ROLB	(R1)	
3779	022416	105421		NEGB	(R1)+	
3780	022420	105441		NEGB	-(R1)	
3781	022422	106111		ROLB	(R1)	
3782	022424	105421		NEGB	(R1)+	
3783	022426	105441		NEGB	-(R1)	
3784	022430	106111		ROLB	(R1)	
3785	022432	105421		NEGB	(R1)+	
3786	022434	105441		NEGB	-(R1)	
3787	022436	106111		ROLB	(R1)	
3788	022440	105421		NEGB	(R1)+	
3789	022442	105441		NEGB	-(R1)	
3790	022444	106111		ROLB	(R1)	
3791	022446	105421		NEGB	(R1)+	
3792	022450	105441		NEGB	-(R1)	
3793	022452	106111		ROLB	(R1)	
3794	022454	005301		DEC	R1	
3795	022456	103401		BCS	1\$	
3796	022460	000000		HALT		:ARITHMETIC OPERATION FAILED RUN DIAG
3797	022462	005367	000022	1\$: DEC	NPRCNT	
3798	022466	001310		BNE	2\$	
3799	022470	104400		HLT		:OPERATION TIMED OUT WAITING FOR INTERRUPT
3800	022472	004567	156566	JSR	R5,PRINT\$	:PRINT MESSAGE
3801	022476	024475		TIMO		
3802	022500	000000		HALT		
3803						
3804	022502	005067	000006	NPRRET: CLR	BCKFLG	
3805	022506	000207		RTS	PC	
3806	022510	000000		NPRCNT: 0		
3807	022512	000000		NPR1: 0		
3808	022514	000000		BCKFLG: 0		
3809						
3810						
3811						:THIS TEST ALLOWS THE OPERATOR TO SPECIFY TWO CYLINDER ADDRESSES
3812						:AND THE PROGRAM WILL THEN SEEK BETWEEN THEM. THE ROUTINE DOES
3813						:NOT CHECK FOR ERRORS.
3814						
3815	022516	000005		CYLSK: RESET		
3816	022520	004567	156556	JSR	R5,PRNTF\$	:FORCE PRINT THE MESSAGE
3817	022524	025333		CON17		

3818	022526	004767	157152		JSR	PC, READ\$	: INPUT MESSAGE
3819	022532	022767	000131	157254	CMP	#131, INPUT\$	: DOES HE WANT HEAD ALIGNMENT ROUTINE
3820	022540	001002			BNE	4\$	: BR IF NO
3821	022542	000167	000424		JMP	HEAD	: GO DO HEAD ALIGNMENT ROUTINE
3822	022546			4\$:			
3823	022546	004567	156530		JSR	R5, PRNTR\$	: FORCE PRINT THE MESSAGE
3824	022552	025566			CON19		
3825	022554	004767	157124		JSR	PC, READ\$	: INPUT MESSAGE
3826	022560	022767	000131	157226	CMP	#131, INPUT\$	: DOES HE WANT HOME SEEKS?
3827	022566	001002			BNE	1\$	: BR IF NO
3828	022570	000167	000526		JMP	HOMERS	: GO DO HOME SEEK ROUTINE
3829	022574			1\$:			
3830	022574	004567	156502		JSR	R5, PRNTR\$	: FORCE PRINT THE MESSAGE
3831	022600	025305			CON15		: CYLINDER A
3832	022602	004767	157076		JSR	PC, READ\$	: INPUT MESSAGE
3833	022606	004767	157250		JSR	PC, PACK\$	: CONVERT INPUT TO A NUMBER
3834	022612	022767	000626	157514	CMP	#626, NUM\$	: IS CYL ADDR TOO HIGH?
3835	022620	101765			BLOS	1\$	: BRANCH IF YES
3836	022622	016767	157506	003116	MOV	NUM\$, CYLA	: SAVE FIRST ADDR
3837	022630			2\$:			
3838	022630	004567	156446		JSR	R5, PRNTR\$	: FORCE PRINT THE MESSAGE
3839	022634	025320			CON16		: CYLINDER B
3840	022636	004767	157042		JSR	PC, READ\$	: INPUT MESSAGE
3841	022642	004767	157214		JSR	PC, PACK\$	: CONVERT INPUT TO A NUMBER
3842	022646	022767	000626	157460	CMP	#626, NUM\$	: IS CYL ADDR TOO HIGH?
3843	022654	101765			BLOS	2\$	: BRANCH IF YES
3844	022656	016767	157452	003064	MOV	NUM\$, CYLB	: SAVE SECOND ADDR
3845	022664			3\$:			
3846	022664	004567	156412		JSR	R5, PRNTR\$	: FORCE PRINT THE MESSAGE
3847	022670	024712			CON4		: DRIVE?
3848	022672	004767	157006		JSR	PC, READ\$	: INPUT MESSAGE
3849	022676	004767	157160		JSR	PC, PACK\$	: CONVERT INPUT TO A NUMBER
3850	022702	022767	000010	157424	CMP	#10, NUM\$	: IS UNIT # TOO HIGH?
3851	022710	101765			BLOS	3\$	: BRANCH IF YES
3852	022712	000241			CLC		
3853	022714	006167	157414		ROL	NUM\$	
3854	022720	006167	157410		ROL	NUM\$	
3855	022724	016767	157404	002732	MOV	NUM\$, FLAG	: SAVE UNIT NO.
3856	022732	004767	174106		JSR	PC, GATTN	: DETERMINE ATTENTION BIT
3857	022736	004567	173674		JSR	R5, DSKNOS	: SELECT THE UNIT
3858	022742	016777	003000	002700	MOV	CYLA, ARPCA	: LOAD THE CYLINDER ADDR
3859	022750	052777	000011	002662	BIS	#11, ARPCS	: ISSUE SEEK COMMAND
3860	022756	105777	002656		TSTB	ARPCS	: WAIT FOR READY
3861	022762	100375			BPL	20\$	
3862	022764	036777	174104	002666	BIT	ATTN, ARPDS	: WAIT FOR ATTENTION
3863	022772	001774			BEQ	1\$	
3864	022774	016777	174074	002656	MOV	ATTN, ARPDS	: CLEAR ATTENTION BIT
3865	023002	005777	002632		TST	ARPCS	: ANY ERRORS?
3866	023006	100022			BPL	2\$	
3867	023010	104400			HLT		: ERROR AFTER SEEK COMMAND
3868	023012	032777	004000	002640	BIT	#B11, ARPDS	: SEEK INCOMPLETE?
3869	023020	001415			BEQ	2\$	: BRANCH IF NO
3870	023022	112777	000015	002610	MOVB	#15, ARPCS	: ISSUE HOME COMMAND
3871	023030	105777	002604		TSTB	ARPCS	: WAIT FOR READY
3872	023034	100375			BPL	21\$	
3873	023036	036777	174032	002614	BIT	ATTN, ARPDS	: WAIT FOR ATTENTION BIT

3874	023044	001774			BEQ	3\$	
3875	023046	016777	174022	002604	MOV	ATTN,ARPD\$	
3876	023054	016777	002670	002566	2\$: MOV	CYLB,ARPCA	;LOAD CYLINDER ADDR
3877	023062	052777	000011	002550	BIS	#11,ARPCS	;ISSUE SEEK COMMAND
3878	023070	105777	002544		22\$: TSTB	ARPCS	;WAIT FOR READY
3879	023074	100375			BPL	22\$	
3880	023076	036777	173772	002554	4\$: BIT	ATTN,ARPD\$	;WAIT FOR ATTENTION
3881	023104	001774			BEQ	4\$	
3882	023106	016777	173762	002544	MOV	ATTN,ARPD\$	;CLEAR ATTENTION
3883	023114	005777	002520		TST	ARPCS	;ANY ERRORS?
3884	023120	100022			BPL	5\$	
3885	023122	104400			HLT		
3886	023124	032777	004000	002526	BIT	#B11,ARPD\$	;SEEK INCOMPLETE?
3887	023132	001415			BEQ	5\$	
3888	023134	112777	000015	002476	MOVB	#15,ARPCS	;ISSUE HOME COMMAND
3889	023142	105777	002472		23\$: TSTB	ARPCS	;WAIT FOR READY
3890	023146	100375			BPL	23\$	
3891	023150	036777	173720	002502	6\$: BIT	ATTN,ARPD\$	;WAIT FOR ATTENTION
3892	023156	001774			BEQ	6\$	
3893	023160	016777	173710	002472	MOV	ATTN,ARPD\$	;CLEAR ATTENTION
3894	023166	000167	177544		5\$: JMP	CYLS1	
3895							
3896							
3897							
3898							
3899							
3900	023172	000005			HEAD: RESET		
3901	023174	004567	156102		JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
3902	023200	025400			CON18		
3903	023202	000000			HALT		
3904	023204	005067	002230		CLR	CON18A	;CLEAR OUT SWITCH DEFINATIONS PRINTOUT
3905	023210	012777	000001	002442	1\$: MOV	#1,ARPD\$	;RESET RP11C CONTROLLER
3906	023216	012777	000222	002424	MOV	#222,ARPCA	;SETUP FOR CYLINDER 146
3907	023224	013746	177570		MOV	#5SWR, -(SP)	
3908	023230	011667	002500		MOV	(SP),WORK	
3909	023234	042767	177600	002472	BIC	#177600,WORK	;CLEAR OUT DRIVE NUMBER
3910	023242	000367	002466		SWAB	WORK	;PUT HEAD IN PROPER POSITION
3911	023246	016777	002462	002376	MOV	WORK,ARPCA	;LOAD HEAD NUMBER INTO DISK ADDRESS
3912	023254	012667	002454		MOV	(SP)+,WORK	;GET DRIVE NUMBER
3913	023260	042767	000377	002446	BIC	#377,WORK	;GE RID OF HEAD ADDRESS
3914	023266	052767	000011	002440	BIS	#11,WORK	;SET A SEEK GO TO DRIVE NUMBER
3915	023274	016777	002434	002336	MOV	WORK,ARPCS	;SET SEEK
3916	023302	032777	110000	002350	2\$: BIT	#B15!B12,ARPD\$	
3917	023310	001774			BEQ	2\$	
3918	023312	105737	177570		3\$: TSTB	#5SWR	;NEW HEAD AND/OR DRIVE
3919	023316	100375			BPL	3\$	;BR IF NO
3920	023320	000733			BR	1\$	;GO TO NEW DRIVE AND HEAD
3921							
3922							
3923							
3924							
3925							
3926							
3927	023322						
3928	023322	004567	155754		HOMERS: JSR	R5,PRNTF\$	;FORCE PRINT THE MESSAGE
3929	023326	024712			CON4		;WHAT DRIVE?

;THIS ROUTINE IS USED FOR HEAD ALIGNMENT - SWITCHES 00-04 IS THE
 ;HEAD TO BE SELECTED, SWITCHES 08-10 IS THE DRIVE TO BE SELECTED -
 ;WHEN SWITCH 07 IS TOGGLED A NEW HEAD AND DRIVE WILL BE SELECTED.

;THIS ROUTINE DOES A HOME SEEK AND THEN TRIES TO DO
 ;A TWO WORD READ FROM CYLINDER 000 WITH OUT A SEEK TO CYLINDER 000
 ;LOOPING BACK TO HOME SEEK

3930	023330	004767	156350		JSR	PC, READS	: INPUT MESSAGE
3931	023334	004767	156522		JSR	PC, PACKS	: CONVERT INPUT TO A NUMBER
3932	023340	022767	000010	156766	CMP	#10, NUMS	: TOO LARGE?
3933	023346	101765			BLOS	HOMERS	: BR IF YES
3934	023350	116777	156760	002264	MOVB	NUMS, ARPCH1	: LOAD UNIT NUMBER INTO CONTROLLER
3935	023356	112777	000015	002254	1\$: MOVB	#15, ARPCH	: ISSUE A HOME SEEK AND GO
3936	023364	105777	002250		2\$: TSTB	ARPCH	: CONTROLLER READY?
3937	023370	100375			BPL	2\$	: BR IF NO
3938	023372	005777	002262		3\$: TST	ARPDS	: DRIVE READY?
3939	023376	100375			BPL	3\$	: BR IF NO
3940	023400	005777	002234		TST	ARPCH	: ERRORS?
3941	023404	100012			BPL	4\$	: BR IF NO ERRORS
3942	023406	104400			HLT		: ERROR WITH HOME SEEKS
3943	023410	032777	001000	002242	BIT	#89, ARPDS	: FILE UNSAFE?
3944	023416	001401			BEQ	5\$	: BR IF NO
3945	023420	000000			HALT		: FILE UNSAFE
3946	023422	012767	023356	155352	5\$: MOV	#1\$, LAD	: SETUP SCOPE LOOP
3947	023430	104000			SCOPE		
3948	023432	005077	002212		4\$: CLR	ARPCA	: SETUP FOR CYLINDER 000
3949	023436	005077	002210		CLR	ARPCA	: SETUP FOR TRACK 00, SECTOR 00
3950	023442	012777	177776	002174	MOV	#-2, ARPWC	: SETUP 2 WORD TRANSFER
3951	023450	012777	025762	002170	MOV	#0, ARPB	: INTO OUTPUT BUFFER
3952	023456	052777	000017	002154	BIS	#17, ARPCH	: READ WITH NO IMPLIED SEEK
3953	023464	105777	002150		6\$: TSTB	ARPCH	: DONE?
3954	023470	100375			BPL	6\$	: BR IF NO
3955	023472	005777	002142		TST	ARPCH	: ERRORS?
3956	023476	100327			BPL	1\$	: BR IF NO ERRORS TO MORE HOME SEEKS
3957	023500	104400			HLT		: HOME SEEK DID NOT RETURN TO CYLINDER 000
3958							: FOR A READ OF ONE WORD
3959	023502	012767	023356	155272	MOV	#1\$, LAD	: SETUP FOR SCOPE LOOP
3960	023510	104000			SCOPE		
3961	023512	000703			BR	HOMERS	: TRY TEST AGAIN WITH NEW DRIVE
3962							
3963							
3964							
3965							
3966							
3967							
3968							
3969	023514	012700	034000		RELOAD: MOV	#34000, R0	: START OF LOADERS
3970	023520	016701	176222		MOV	MEMSIZ, R1	: TOP OF MEMORY
3971	023524	042701	003777		BIC	#3777, R1	: MAKE IT A 1K TRANSFER
3972	023530	012737	023550	000004	MOV	#1\$, ARERRVEC	: SET UP FOR TRAP
3973	023536	012737	000340	000006	MOV	#340, ARERRVEC+2	
3974	023544	012021			2\$: MOV	(R0)+, (R1)+	: RESTORE LOADER
3975	023546	000776			BR	2\$	
3976	023550	012737	000006	000004	1\$: MOV	#6, ARERRVEC	: RESTORE TRAPCATCHER
3977	023556	005037	000006		CLR	ARERRVEC+2	
3978	023562	005037	000176		CLR	#176	
3979	023566	005067	002164		CLR	LDRFLG	: SET LOADER FLAG TO LOADERS AT TOP OF MEMORY
3980	023572	012707	000176		MOV	#176, PC	: FINISHED
3981							
3982							
3983							
3984							
3985	023576	005015	047105	020104	MES1: .EVEN	.ASCIZ	<15><12>/END OF PASS /

3986	023604	043117	050040	051501		
3987	023612	020123	000			
3988	023615	015	051012	042520	MES1A:	.ASCIZ <15><12>/RPER= /
3989	023622	036522	000040			
3990						
3991	023626	005015	050122	051503	MES2:	.ASCIZ <15><12>/RPCS= /
3992	023634	020075	000			
3993						
3994	023637	015	051012	042120	MES2A:	.ASCIZ <15><12>/RPDS= /
3995	023644	036523	000040			
3996	023650	005015	050122	040503	MES2B:	.ASCIZ <15><12>/RPCA= /
3997	023656	020075	000			
3998	023661	015	051012	042120	MES2C:	.ASCIZ <15><12>/RPDA= /
3999	023666	036501	000040			
4000	023672	005015	052523	040503	MES2D:	.ASCIZ <15><12>/SUCA= /
4001	023700	020075	000			
4002						
4003	023703	015	046412	046505	MES4:	.ASCIZ <15><12>/MEM ADDR= /
4004	023710	040440	042104	036522		
4005	023716	000040				
4006						
4007	023720	005015	040527	052111	MES5:	.ASCIZ <15><12>/WAIT 5 SEC. AND TURN OFF PDP-11 PWR/
4008	023726	032440	051440	041505		
4009	023734	020056	047101	020104		
4010	023742	052524	047122	047440		
4011	023750	043106	050040	050104		
4012	023756	030455	020061	053520		
4013	023764	000122				
4014						
4015	023766	005015	042524	052123	MES6:	.ASCIZ <15><12>/TEST NO /
4016	023774	047040	020117	000		
4017						
4018	024001	015	052012	052117	MES7:	.ASCIZ <15><12>/TOT REREADS ON ERR= /
4019	024006	051040	051105	040505		
4020	024014	051504	047440	020116		
4021	024022	051105	036522	000040		
4022						
4023						
4024	024030	005015	052123	052101	MES8:	.ASCIZ <15><12>/STAT ERR/
4025	024036	042440	051122	000		
4026						
4027	024043	015	053412	020104	MES9:	.ASCIZ <15><12>/WD CNT INTO SECT= /
4028	024050	047103	020124	047111		
4029	024056	047524	051440	041505		
4030	024064	036524	000040			
4031						
4032	024070	005015	040520	020124	MES10:	.ASCIZ <15><12>/PAT IN USE= /
4033	024076	047111	052440	042523		
4034	024104	020075	000			
4035						
4036	024107	015	052412	044516	MES11:	.ASCIZ <15><12>/UNIT NO. /
4037	024114	020124	047516	020056		
4038	024122	000				
4039						
4040	024123	103	046131	000	MES12:	.ASCIZ /CYL/
4041						

MO6

4042	024127	015	051012	040505	MES13: .ASCIZ <15><12>/READ NO. /
4043	024134	020104	047516	020056	
4044	024142	000			
4045					
4046	024143	015	041412	046131	MES14: .ASCIZ <15><12>/CYL= /
4047	024150	020075	000		
4048					
4049	024153	015	044012	040505	MES15: .ASCIZ <15><12>/HEAD= /
4050	024160	036504	000040		
4051					
4052	024164	005015	042523	052103	MES16: .ASCIZ <15><12>/SECT= /
4053	024172	020075	000		
4054					
4055	024175	015	041412	046517	MES17: .ASCIZ <15><12>/COMP ERR/
4056	024202	020120	051105	000122	
4057					
4058	024210	005015	054105	042520	MES18: .ASCIZ <15><12>/EXPECTED /
4059	024216	052103	042105	000040	
4060					
4061	024224	005015	042522	053103	MES19: .ASCIZ <15><12>/RECVD /
4062	024232	020104	020040	000040	
4063	024240	005015	047516	052440	MES20: .ASCIZ <15><12>/NO UNITS AVAIL/
4064	024246	044516	051524	040440	
4065	024254	040526	046111	000	
4066	024261	015	051012	040505	MES21: .ASCIZ <15><12>/READ CNTR = /
4067	024266	020104	047103	051124	
4068	024274	036440	020040	000	
4069	024301	015	046012	051104	MES22: .ASCII <15><12>/LDRS MOVED TO 8K/
4070	024306	020123	047515	042526	
4071	024314	020104	047524	034040	
4072	024322	113			
4073	024323	015	052012	020117	.ASCIZ <15><12>/TO REST LDR START AT LOC /
4074	024330	042522	052123	046040	
4075	024336	051104	051440	040524	
4076	024344	052122	040440	020124	
4077	024352	047514	020103	000040	
4078	024360	005015	044513	040520	MES23: .ASCIZ <15><12>/KIPAR2 = /
4079	024366	031122	036440	020040	
4080	024374	000			
4081	024375	040	020040	020040	MES24: .ASCIZ / KIPAR3 = /
4082	024402	020040	045440	050111	
4083	024410	051101	020063	020075	
4084	024416	000040			
4085	024420	005015	044513	040520	MES25: .ASCIZ <15><12>/KIPAR4 = /
4086	024426	032122	036440	020040	
4087	024434	000			
4088	024435	040	020040	020040	MES26: .ASCIZ / KIPAR5 = /
4089	024442	020040	045440	050111	
4090	024450	051101	020065	020075	
4091	024456	000040			
4092	024460	005015	044513	040520	MES27: .ASCIZ <15><12>/KIPAR6 = /
4093	024466	033122	036440	020040	
4094	024474	000			
4095					
4096					
4097	024475	015	050012	047522	TIM0: .ASCIZ <15><12>/PROC BCKGRD TEST TIMED OUT/

4098	024502	020103	041502	043513	
4099	024510	042122	052040	051505	
4100	024516	020124	044524	042515	
4101	024524	020104	052517	000124	
4102					: CONVERSATION TEXT
4103					:
4104					:
4105	024532	005015	052123	040504	SPECMES: .ASCIZ <15><12>/STDARD WDS XFERRED= /
4106	024540	042122	053440	051504	
4107	024546	054040	042506	051122	
4108	024554	042105	020075	000	
4109					
4110	024561	015	042012	052101	CON1: .ASCIZ <15><12>/DATA TEST ONLY? (Y OR N)/
4111	024566	020101	042524	052123	
4112	024574	047440	046116	037531	
4113	024602	024040	020131	051060	
4114	024610	047040	000051		
4115					
4116	024614	005015	052515	052114	CON2: .ASCIZ <15><12>/MULTI DRIVE MODE?(Y OR N)/
4117	024622	020111	051104	053111	
4118	024630	020105	047515	042504	
4119	024636	024077	020131	051117	
4120	024644	047040	000051		
4121					
4122	024650	005015	052516	041115	CON3: .ASCIZ <15><12>/NUMBER OF DRIVES 1 TO 10 OCTAL?/
4123	024656	051105	047440	020106	
4124	024664	051104	053111	051505	
4125	024672	030440	052040	020117	
4126	024700	030061	047440	052103	
4127	024706	046101	000077		
4128					
4129	024712	005015	044127	041511	CON4: .ASCIZ <15><12>/WHICH DRIVE?/
4130	024720	020110	051104	053111	
4131	024726	037505	000		
4132					
4133	024731	015	047412	052120	CON5: .ASCIZ <15><12>/OPTIONAL WORD COUNT? (Y OR N)/
4134	024736	047511	040516	020114	
4135	024744	047527	042122	041440	
4136	024752	052517	052116	020077	
4137	024760	054450	047440	020122	
4138	024766	024516	000		
4139					
4140	024771	015	046012	047105	CON6: .ASCIZ <15><12>/LENGTH? (1 TO SWRDCT)/
4141	024776	052107	037510	024040	
4142	025004	020061	047524	051440	
4143	025012	051127	041504	024524	
4144	025020	000			
4145					
4146	025021	015	051412	040524	CON7: .ASCIZ <15><12>/STARTING HEAD?/
4147	025026	052122	047111	020107	
4148	025034	042510	042101	000077	
4149					
4150	025042	005015	047504	054440	CON7A: .ASCIZ <15><12>/DO YOU WISH TO SELECT THE DISK TEST ADDR?(Y OR N)/
4151	025050	052517	053440	051511	
4152	025056	020110	047524	051440	
4153	025064	046105	041505	020124	

4154	025072	044124	020105	044504	
4155	025100	045523	052040	051505	
4156	025106	020124	042101	051104	
4157	025114	024077	020131	051117	
4158	025122	047040	000051		
4159					
4160	025126	005015	052123	051101	CON7B: .ASCIZ <15><12>/STARTING SECT?/
4161	025134	044524	043516	051440	
4162	025142	041505	037524	000	
4163					
4164	025147	015	051412	040524	CON7C: .ASCIZ <15><12>/STARTING CYL?/
4165	025154	052122	047111	020107	
4166	025162	054503	037514	000	
4167					
4168	025167	015	047412	052120	CON8: .ASCIZ <15><12>/OPTAL DATA PAN NO.?/
4169	025174	046101	042040	052101	
4170	025202	020101	040520	020116	
4171	025210	047516	037456	000	
4172					
4173	025215	015	053412	044522	CON9: .ASCIZ <15><12>/WRITE?(Y OR N)/
4174	025222	042524	024077	020131	
4175	025230	051117	047040	000051	
4176					
4177	025236	005015	051127	052111	CON10: .ASCIZ <15><12>/WRITE CHECK?(Y OR N)/
4178	025244	020105	044103	041505	
4179	025252	037513	054450	047440	
4180	025260	020122	024516	000	
4181					
4182	025265	015	051012	040505	CON11: .ASCIZ <15><12>/READ?(Y OR N)/
4183	025272	037504	054450	047440	
4184	025300	020122	024516	000	
4185					
4186	025305	015	041412	046131	CON15: .ASCIZ <15><12>/CYL "A"?/
4187	025312	021040	021101	000077	
4188					
4189	025320	005015	054503	020114	CON16: .ASCIZ <15><12>/CYL "B"?/
4190	025326	041042	037442	000	
4191	025333	015	053412	047101	CON17: .ASCIZ <15><12>/WANT HEAD ALIGN ROUTINE?(Y OR N) /
4192	025340	020124	042510	042101	
4193	025346	040440	044514	047107	
4194	025354	051040	052517	044524	
4195	025362	042516	024077	020131	
4196	025370	051117	047040	020051	
4197	025376	000040			
4198	025400	005015	042523	020124	CON18: .ASCII <15><12>/SET SWCH THEN PRESS CONTINUE./
4199	025406	053523	044103	052040	
4200	025414	042510	020116	051120	
4201	025422	051505	020123	047503	
4202	025430	052116	047111	042525	
4203	025436	056			
4204		025440			.EVEN
4205	025440	005015	053523	044103	CON18A: .ASCII <15><12>/SWCH 00-04 = HEAD/
4206	025446	030040	026460	032060	
4207	025454	036440	044040	040505	
4208	025462	104			
4209	025463	015	051412	041527	.ASCII <15><12>/SWCH 08-10 = DRIVE/

4210	025470	020110	034060	030455	
4211	025476	020060	020075	051104	
4212	025504	053111	105		
4213	025507	015	052012	043517	.ASCIZ <15><12>'TOGGLE SW 07 TO SELECT NEW HEAD AND/OR DRIVE'
4214	025514	046107	020105	053523	
4215	025522	030040	020067	047524	
4216	025530	051440	046105	041505	
4217	025536	020124	042516	020127	
4218	025544	042510	042101	040440	
4219	025552	042116	047457	020122	
4220	025560	051104	053111	000105	
4221	025566	005015	040527	052116	CON19: .ASCIZ <15><12>'WANT CONTINUOUS HOME SEEKS? '
4222	025574	041440	047117	044524	
4223	025602	052516	052517	020123	
4224	025610	047510	042515	051440	
4225	025616	042505	051513	020077	
4226	025624	000			
4227					
4228	025625	015	042412	042116	END: .ASCIZ <15><12>'END/
4229	025632	000			
4230		025634			.EVEN
4231					:
4232					:
4233					:
4234					:
4235					:DISK I/O REGISTERS
4236					
4237	025634	000254			VECTOR: 254
4238	025636	000256			STATUS: 256
4239	025640	176714			RPCS: 176714
4240	025642	176715			RPCS1: 176715
4241	025644	176716			RPWC: 176716
4242	025646	176720			RPBA: 176720
4243	025650	176722			RPCA: 176722
4244	025652	176724			RPDA: 176724
4245	025654	176725			RPDA1: 176725
4246	025656	176712			RPER: 176712
4247	025660	176710			RPDS: 176710
4248	025662	176734			SUCA: 176734
4249					
4250					
4251					
4252					:DEDICATED REGISTERS
4253	025664	000000			FLAG: 0
4254	025666	000000			SCYL: 0
4255	025670	000000			SHED: 0
4256	025672	000000			SSEC: 0
4257	025674	000000			WRDCT: 0
4258	025676	000000			CYLINDER: 0
4259	025700	000000			DMA: 0
4260	025702	000000			SWITCH: 0
4261	025704	000000			PATNU: 0
4262	025706	000000			BUF: 0
4263	025710	000000			SWRDCT: 0
4264	025712	000000			SAVE: 0
4265	025714	000000			DSKNOR: 0

:DISK INTERRUPT TRAP LOCATION
 :INTERUPT PRIORITY ASSIGNMENT
 :DISK-CONTROL REGISTER
 :UPPER BYTE OF CONTROL REGISTER
 :WORD COUNT REGISTER
 :CURRENT ADDR REGISTER
 :CYLINDER ADDR REGISTER
 :DISK ADDR REGISTER
 :TRACK ADDRESS
 :ERROR REGISTER
 :DEVICE STATUS REGISTER
 :SELECTED UNIT CYLINDER ADDR REG

:INTERNAL PROGRAM FLAG WORD
 :OPERATOR SELECTED CYLINDER
 :OPERATOR SELECTED HEAD
 :OPERATOR SELECTED SECTOR
 :WORKING WORD COUNT
 :WORKING CYLINDER ADDR
 :WORKING DISK ADDR

:DATA PATTERN INDEX
 :WORKING DATA BUFFER
 :STANDARD WORD COUNT

:MAXIMUM UNIT NUMBER

4266	025716	000000	BLOCK:	0	
4267	025720	000000	PASSC:	0	
4268	025722	000000	INBUF:	0	:CONTAINS START OF INPUT BUFFER
4269	025724	000000	RDERR:	0	;READ RETRY COUNTER
4270	025726	000000	ACNVX:	0	
4271	025730	000000	TESTNO:	0	
4272	025732	000000	ERCOUNT:	0	
4273	025734	000000	WORK:	0	
4274	025736	000000	WORK1:	0	
4275	025740	000000	WORK2:	0	
4276	025742	000000	WORK3:	0	
4277	025744	000000	WORK4:	0	
4278	025746	000000	CYLA:	0	
4279	025750	000000	CYLB:	0	
4280	025752	000000	SEEK1:	0	
4281	025754	000000	CNTA:	0	
4282	025756	000000	LDRFLG:	0	;WHEN =0 LOADERS ARE AT TOP OF MEMORY
4283					
4284					
4285			.EVEN		
4286	025760	005015	HEADER:	.ASCII <15><12>	
4287	025762	042115	OUTBUF:	.ASCII 'MD-11-DZRPB-E DISK RELIABILITY TEST'	
4288	025770	055104			
4289	025776	020105			
4290	026004	045523			
4291	026012	040511			
4292	026020	054524			
4293	026026	124			
4294	026027	015		.ASCII <15><12><12>'SWITCH OPTIONS'	
4295	026034	052111			
4296	026042	052120			
4297	026050	005015		.ASCII <15><12>'SW <15> =1....HALT ON ERROR'	
4298	026056	032461			
4299	026064	027056			
4300	026072	052114			
4301	026100	051105			
4302	026105	015		.ASCII <15><12>'SW <14> =1....LOOP ON ERROR'	
4303	026112	030474			
4304	026120	027061			
4305	026126	047517			
4306	026134	042440			
4307	026142	005015		.ASCII <15><12>'SW <13> =1....INHIBIT PRINTOUTS'	
4308	026150	031461			
4309	026156	027056			
4310	026164	044510			
4311	026172	051120			
4312	026200	052125			
4313	026203	015		.ASCII <15><12>'SW <12> =1....INHIBIT BACKGROUND TEST'	
4314	026210	030474			
4315	026216	027061			
4316	026224	044116			
4317	026232	041040			
4318	026240	052517			
4319	026246	051505			
4320	026251	015		.ASCII <15><12>'SW <11> =1....RING BELL ON ERROR'	
4321	026256	030474			

4322	026264	027061	027056	051056
4323	026272	047111	020107	042502
4324	026300	046114	047440	020116
4325	026306	051105	047522	122
4326	026313	015	051412	020127
4327	026320	030474	037060	036440
4328	026326	027061	027056	046056
4329	026334	047517	020120	047117
4330	026342	052040	051505	124
4331	026347	015	051412	020127
4332	026354	030074	037071	036440
4333	026362	027061	027056	044456
4334	026370	044116	041111	052111
4335	026376	042040	052101	020101
4336	026404	047503	050115	051101
4337	026412	051511	047511	116
4338	026417	015	051412	020127
4339	026424	030074	037070	036440
4340	026432	027061	027056	042456
4341	026440	052116	051105	041440
4342	026446	047117	042526	051522
4343	026454	052101	047511	020116
4344	026462	047515	042504	
4345	026466	005015	020040	020040
4346	026474	020040	020040	020040
4347	026502	020040	020040	042524
4348	026510	052123	051440	046105
4349	026516	041505	042524	020104
4350	026524	020060	020055	020062
4351	026532	044507	042526	020123
4352	026540	042101	051104	051505
4353	026546	020123	047503	053116
4354	026554	051105	040523	044524
4355	026562	047117		
4356	026564	005015	020040	020040
4357	026572	020040	020040	020040
4358	026600	020040	020040	042524
4359	026606	052123	051440	046105
4360	026614	041505	042524	020104
4361	026622	020063	020055	020067
4362	026630	044507	042526	020123
4363	026636	040504	040524	041440
4364	026644	047117	042526	051522
4365	026652	052101	047511	116
4366	026657	015	051412	020127
4367	026664	030074	037067	036440
4368	026672	027061	027056	044456
4369	026700	044116	041111	052111
4370	026706	051040	047101	047504
4371	026714	020115	042523	045505
4372	026722	020123	052504	044522
4373	026730	043516	042040	052101
4374	026736	020101	042524	052123
4375	026744	005015	053523	036040
4376	026752	033060	020076	030475
4377	026760	027056	027056	047111

.ASCII <15><12>'SW <10> =1....LOOP ON TEST'

.ASCII <15><12>'SW <09> =1....INHIBIT DATA COMPARISION'

.ASCII <15><12>'SW <08> =1....ENTER CONVERSATION MODE'

.ASCII <15><12>'

TEST SELECTED 0 - 2 GIVES ADDRESS CONVERSATION'

.ASCII <15><12>'

TEST SELECTED 3 - 7 GIVES DATA CONVERSATION'

.ASCII <15><12>'SW <07> =1....INHIBIT RANDOM SEEKS DURING DATA TEST'

.ASCII <15><12>'SW <06> =1....INHIBIT USE OF MEMORY MANAGEMENT'

4378	026766	044510	044502	020124	
4379	026774	051525	020105	043117	
4380	027002	046440	046505	051117	
4381	027010	020131	040515	040516	
4382	027016	042507	042515	052116	
4383	027024	005015	053523	036040	.ASCII <15><12>'SW <05>.....COMPARISION ERRORS PRINTOUT'
4384	027032	032460	027076	027056	
4385	027040	027056	027056	047503	
4386	027046	050115	051101	051511	
4387	027054	047511	020116	051105	
4388	027062	047522	051522	050040	
4389	027070	044522	052116	052517	
4390	027076	124			
4391	027077	015	051412	020127	.ASCII <15><12>'SW <04>.....REREAD INFORMATION'
4392	027104	030074	037064	027056	
4393	027112	027056	027056	051056	
4394	027120	051105	040505	020104	
4395	027126	047111	047506	046522	
4396	027134	052101	047511	116	
4397	027141	015	051412	020127	.ASCIZ <15><12>'SW <03> =1....SELECT TEST IN SWCH 00 TO 02'
4398	027146	030074	037063	036440	
4399	027154	027061	027056	051456	
4400	027162	046105	041505	020124	
4401	027170	042524	052123	044440	
4402	027176	020116	053523	044103	
4403	027204	030040	020060	047524	
4404	027212	030040	000062		
4405	027216	005015	025012	020052	MES28: .ASCIZ <15><12><12>'** MAP OF RP DRIVES **/<15><12>
4406	027224	040515	020120	043117	
4407	027232	051040	020120	051104	
4408	027240	053111	051505	025040	
4409	027246	006452	000012		
4410	027252	020072	050122	031060	MES29: .ASCIZ /: RP02/
4411	027260	000			
4412	027261	072	051040	030120	MES30: .ASCIZ /: RP03/
4413	027266	000063			
4414	027270	020072	043117	026506	MES31: .ASCIZ /: OFF-LINE/
4415	027276	044514	042516	000	
4416	027303	072	047040	052117	MES32: .ASCIZ /: NOT READY/
4417	027310	051040	040505	054504	
4418	027316	000			
4419	027317	015	005012	043111	MES33: .ASCII <15><12><12>'IF THE DRIVE TYPES SEEN BY THE SYSTEM/
4420	027324	052040	042510	042040	
4421	027332	044522	042526	052040	
4422	027340	050131	051505	051440	
4423	027346	042505	020116	054502	
4424	027354	052040	042510	051440	
4425	027362	051531	042524	115	
4426	027367	015	040412	042522	.ASCII <15><12>'ARE INCORRECT, THEN BIT 13 (SU RP03)/
4427	027374	044440	041516	051117	
4428	027402	042522	052103	020054	
4429	027410	044124	047105	041040	
4430	027416	052111	030440	020063	
4431	027424	051450	020125	050122	
4432	027432	031460	051		
4433	027435	015	047412	020106	.ASCII <15><12>'OF THE RP11C'S DEVICE STATUS REGISTER/

4434	027442	044124	020105	050122		
4435	027450	030461	023503	020123		
4436	027456	042504	044526	042503		
4437	027464	051440	040524	052524		
4438	027472	020123	042522	044507		
4439	027500	052123	051105			
4440	027504	005015	040515	020131	.ASCII <15><12>/MAY BE FAULTY./	
4441	027512	042502	043040	052501		
4442	027520	052114	027131			
4443	027524	005015	000012		.ASCIZ <15><12><12>	
4444						
4445						
4446						
4447					:ROUTINE TO CHECK FOR DRIVE STATUS AND REPORT TO	
4448	027530	004567	151546		:PROCESSOR CONSOLE TTY.	
4449	027534	027216		SUBSIZ:	JSR R5,PRNTFS ;FORCE PRINT THE MESSAGE	
4450	027536	005000			MES28 ;< ** MAP OF RP DRIVES ** >.	
4451	027540	004567	151536	1\$:	CLR R0 ;CLEAR COUNTER FOR UNIT NO.	
4452	027544	024107			JSR R5,PRNTFS ;FORCE PRINT THE MESSAGE	
4453	027546	010067	152010		MES11 ;<UNIT NO.>	
4454	027552	004767	151622		MOV R0,TTY	
4455	027556	110077	176060		JSR PC,PRINTB ;FORCE TYPE LOCATION - SUPPRESS ZEROS	
4456	027562	032777	040000	176070	MOVB R0,ARPCSI ;CHECK THE UNIT SELCTED BY R0	
4457	027570	001004			BIT #B14,ARPCSI ;IS IT ON-LINE?	
4458	027572	004567	151504		BNE 3\$;IF YES THEN PROCEED ON, ELSE	
4459	027576	027270			JSR R5,PRNTFS ;REPORT THAT THE SELECTED DRIVE	
4460	027600	000423			MES31 ;IS OFF-LINE.	
4461	027602	032777	100000	176050	3\$:	BR 6\$;GO CHECK THE NEXT DRIVE.
4462	027610	001004			BIT #B15,ARPCSI ;IS THE SELECTED DRIVE READY?	
4463	027612	004567	151464		BNE 4\$;IF YES THEN PROCEED ON, ELSE	
4464	027616	027303			JSR R5,PRNTFS ;REPORT THAT THE SELECTED DRIVE IS	
4465	027620	000413			MES32 ;NOT READY!	
4466	027622	032777	020000	176030	4\$:	BR 6\$;GO CHECK THE NEXT DRIVE.
4467	027630	001004			BIT #B13,ARPCSI ;IS THE SELECTED DRIVE AN RP03?	
4468	027632	004567	151444		BNE 5\$;IF YES THEN PROCEED ON, ELSE	
4469	027636	027252			JSR R5,PRNTFS ;REPORT THAT THE SELECTED DRIVE	
4470	027640	000403			MES29 ;IS AN RP02.	
4471	027642	004567	151434	5\$:	BR 6\$;GO CHECK THE NEXT DRIVE.	
4472	027646	027261			JSR R5,PRNTFS ;REPORT THAT THE SELECTED DRIVE	
4473	027650	005200		6\$:	MES30 ;IS AN RP03.	
4474	027652	032700	000010		INC R0 ;UPDATE DRIVE UNIT NO.	
4475	027656	001730			BIT #B3,R0 ;CHECKED ALL THE DRIVE(S) YET?	
4476	027660	004567	151416		BEQ 1\$;BRANCH IF NO, ELSE PROCEED ONWARD	
4477	027664	027317			JSR R5,PRNTFS ;FORCE PRINT THE MESSAGE	
4478					MES33 ;<IF THE DRIVE TYPES SEEN BY THE SYSTEM>	
4479	027666	000207			RTS PC ;<ARE INCORRECT, THEN RPDS BIT 13 MAY BE FAULTY.>	
4480						;RETURN TO CALLER.
4481						
4482		000001			.END	

ACNVX	025726	1373*	1374*	1375*	1376*	1377	3481*	3482*	3485	3487*	3490	3494*	3495*	3496
ADAM	020344	4270#												
ADERC	005444	3429	3437#	3507										
ADERC1	005462	1637	1649#											
ADERR	005454	1650	1652#											
ADHGT	005370	1641	1651#											
ADHGT1	005374	1629	1632	1634#										
ADTER1	005504	1635#	1648											
ADTST	003560	1633	1646	1655	1658#									
ADT1	004036	1207	1348#	2824										
ADT2	004514	1387	1390	1409#										
ADT3	005576	1391	1488	1508#										
ADT32	006024	1392	1671	1681#										
ASKWC	003064	1719#	1741	1748	1753									
ATTN	017074	1246	1258#											
ATTNB	017076	1455	3211*	3213*	3217#	3862	3864	3873	3875	3880	3882	3891	3893	
BCKFLG	022514	3213	3218#											
BELL	001250	3003	3738#	3804*	3808#									
BLAD1	020312	965	986#											
BLOCK	025716	3430#												
BLSZ	016764	3094	3100*	3193*	3195	3197	3198*	3199	3201*	3451*	3456*	3461	3463*	4266#
BUF	025706	3093	3193#											
		1541*	1692*	1719*	1780*	1927*	2025*	2154*	2219*	2324*	2372*	2413*	2510*	2691*
BUFEXI	016402	2694*	2851*	2930	3715*	4262#								
BUFINX	016434	3092	3113	3123#										
B0	= 000001	3122	3129#											
B1	= 000002	840#	2090	2964	3326	3520	3548	3629						
B10	= 002000	839#	1277	1955	2953	3289	3329	3725						
B11	= 004000	830#	1224	1386	1434	1487	1670	1754	2002	2180	2460	2644	2788	2996
		829#	962	1230	1361	1369	1468	1602	1659	1706	1734	2137	2800	3048
		3868	3886											
B12	= 010000	828#	1336	2078	2517	2531	2540	2558	2698	2707	2745	3916		
B13	= 020000	827#	966	992	1010	1013	1343	2526	4466					
B14	= 040000	826#	926	1329	1631	1892	2048	2082	2242	2296	2333	2511	2752	4456
B15	= 100000	825#	979	1319	1473	1481	1607	1664	1711	1739	1747	1849	1959	2009
		2086	2122	2142	2337	2860	3916	4461						
B2	= 000004	838#	2329	3333										
B3	= 000010	837#	1379	1824	4474									
B4	= 000020	836#	2382	2395	2422	2440	2950	3444						
B5	= 000040	835#	1284	2405	2426	2450	2633	3076	3090	3502				
B6	= 000100	834#	1179	1192	2166	2205	2320	2464	2817	3042	3446	3517	3560	
B7	= 000200	833#												
B8	= 000400	832#	1204	2262	3573									
B9	= 001000	831#	2316	2556	2624	2750	2993	3050	3716	3724	3727	3943		
C	= 000001	790#												
CHKMOD	003550	1342	1344#											
CLRTST	010360	2150#	2169											
CMDAE	016374	3101	3121#											
CMPLP	020220	3404	3415#	3434										
CMPLP1	020262	3406	3424#											
CMPLP2	020164	3407#	3435											
CNTA	025754	1203*	2489*	2549*	2610*	2619*	2620	2714*	2772*	2781*	2782	3324	4281#	
COMPAR	020064	2590	2754	3394#										
CON1	024561	1220	4110#											
CON10	025236	1339	4177#											
CON11	025265	1332	4182#											

CON15	025305	3831	4186#															
CON16	025320	3839	4189#															
CON17	025333	3817	4191#															
CON18	025400	3902	4198#															
CON18A	025440	3904*	4205#															
CON19	025566	3824	4221#															
CON2	024614	1226	4116#															
CON3	024650	1233	4122#															
CON4	024712	1249	3847	3929	4129#													
CON5	024731	1260	4133#															
CON6	024771	1266	4140#															
CON7	025021	1295	4146#															
CON7A	025042	1280	4150#															
CON7B	025126	1303	4160#															
CON7C	025147	1287	4164#															
CON8	025167	1312	4168#															
CON9	025215	1325	4173#															
CYLA	025746	1442*	1443*	1444*	1445*	1527*	1528*	1529*	1530*	2069*	2070*	2071*	2072*	3836*				
		3858	4278#															
CYLB	025750	3844*	3876	4279#														
CYLIND	025676	1200*	1417*	1422	1459	1461	1480*	1483*	1543*	1546	1591	1636	1649	1690*				
		1693	1702	1704*	1715*	1728	1730	1746*	1751*	1776*	1925*	1942*	1965*	1991*				
		2023*	2152*	2221*	2274*	2495*	2682*	2799*	2848*	2929	3025	3079*	3111*	3112				
		3124*	3131	3460	4258#													
CYLSK	022516	1213	3815#															
CYLSI	022736	3857#	3894															
DATAT	012330	1388	1395	2461	2484#													
DATP	012466	2502	2507#	2635														
DATTES	003016	1229	1247*	1253														
DECBLK	016302	3100#	3109	3114														
DEC CY	015464	2971	2975#															
DECTK	015440	2967	2970#															
DELME\$	015370	2954	2961#															
DISBUF	016226	1611	1668	2524	2538	2629	3090#											
DISPLA=	177570	819#																
DKINT	015272	2499	2843#	3185														
DKII	015576	2951	2993#															
DMA	025700	1201*	1542*	1548	1551													

ESH	012720	3972*	3973*	3976*	3977*										
EXGEN	017352	2550#	2617	2621											
EXIT	021222	3277	3282#												
EXMFLG	003752	3554#	3649												
EXP\$	020024	1370	1379#												
		1461*	1575*	1579*	1580	1582*	1585*	1586	1591*	1594*	1595	1597*	1598	1649*	
		1651*	1730*	1897*	1909*	1977*	2050*	2052	2253*	2309*	2401*	2446*	2888*	2898*	
		3358	3365#	3409*	3413*	3418*	3422*	3425*	3426						
EXREF	021046	3527#	3532												
EXTDR	016634	3128	3135	3161#											
EXTEND	012246	2457#	2470												
EXTMEN	020766	1196	3517#												
EXTPP	014556	2809	2816	2824#											
EXTRP	012274	2368	2456	2464#											
EXTTRP	012310	2346	2465	2467#											
EXTTST	011502	2346#													
EXTTI	012030	2410#													
FILDAT	017154	3239#	3241												
FLAG	025664	1178*	1192*	1206*	1224*	1230*	1257*	1277*	1284*	1319*	1323*	1329*	1336*	1343*	
		1344	1361*	1369	1373	1386	2166	2171	2200	2205	2262	2316	2320	2464	
		2501	2508	2511	2526	2540	2543	2567	2622	2624	2633	2636	2660	2671	
		2692	2723	2800	2802	2806*	2808*	2817	2877	3076	3090	3121	3126*	3160*	
		3166	3208	3232	3397	3446	3517	3560	3573*	3716*	3724*	3725	3727	3855*	
		4253#													
FNCT	015270	2933*	2935*	2936	2938#										
FUNCT	015202	1556	1624	1694	1720	1787	1795	1820	1867	1876	1885	1933	1951	1966	
		1994	2031	2040	2063	2101	2155	2222	2235	2275	2289	2325	2373	2387	
		2414	2432	2515	2529	2554	2696	2705	2718	2856	2867	2927#			
GATTN	017044	1454	3208#	3856											
HEAD	023172	3821	3900#												
HEADER	025760	1174	1175	1195*	4286#										
HINUM	001700	1054	1064	1075*	1079#	2678	3402*	3422							
HINUN	017356	3252	3254	3264	3278*	3284#									
HISAV	017362	3252*	3286#	3402											
HLT =	104400	866#	1430	1436	1448	1452	1457	1463	1467	1521	1533	1537	1564	1568	
		1576	1588	1600	1630	1653	1700	1726	1732	1793	1801	1826	1834	1873	
		1882	1891	1899	1910	1939	1957	1961	1972	1978	2004	2010	2037	2046	
		2059	2075	2080	2084	2088	2092	2107	2134	2163	2228	2241	2255	2281	
		2295	2311	2331	2335	2339	2380	2384	2393	2397	2403	2407	2420	2424	
		2428	2438	2442	2448	2452	2862	2890	2900	2952	3047	3175	3449	3450	
		3799	3867	3885	3942	3957									
HLTADS	001244	970*	971*	972	973	984#									
HLTCTS	001242	972*	983#	3289	3326	3329	3333								
HNFCK	007542	2022#	2061												
HNF1	007752	2063#	2093												
HNF2	010110	2095#	2108												
HOMERS	023322	3828	3927#	3933	3961										
ICNT	001000	916#	1198*	2810*	2813										
INBUF	025722	1719	1728	1731	3549*	3550*	4268#								
INCSEC	016252	3095#	3102												
INCSUR	016312	3098	3103#												
INDRVE	014472	2805	2808#												
INIT	016710	1172	1687	1774	2204	2659	3181#								
INPUT\$	002014	1082	1083	1106#	1117	1158	1222	1228	1262	1282	1327	1334	1341	3819	
		3826													
INTCK	004502	1414	1492#												

[illegible]

RANDOM	017214	2674	3236	3251#										
RANDS	001564	1051#	2676	3417										
RANEX	013424	1396	2645	2653#										
RANLOP	013700	2691#	2703											
RDATAT	012366	2491#	2642	2646										
RDERR	025724	2550*	2595*	2596	2598	2606	2608	2614	2616*	2715*	2757*	2758	2760	2768
		2770	2776	2778*	2948	2957	3442	3500	4269#					
RDSECT	005314	1619#	1666	1669										
READLS	002060	1102	1108#											
READMS	002054	1093	1107#											
READS	001704	1081#	1221	1227	1234	1250	1261	1267	1281	1288	1296	1304	1313	1326
		1333	1340	3818	3825	3832	3840	3848	3930					
READ1	012714	2544	2549#	2632										
RECS	020026	1462*	1571*	1572*	1573	1578*	1586	1590*	1598	1652*	1731*	1898*	1908*	1974*
		1975	2058*	2254*	2310*	2402*	2447*	2889*	2899*	3362	3366#	3440*		
REDAC	015476	2965	2969	2974	2977#									
RELOAD	023514	3647	3969#											
REPOEN	014502	2801	2807	2810#										
RESTR	002370	1176	1178#											
RESTS	001064	947#	1076	1152										
RESVEC=	000010	805#												
RMEMT	010632	2204#	2459	2462										
RMMEMT	011106	2206	2262#											
RORBLK	017030	3196	3200#											
RPBA	025646	2930*	3951*	4242#										
RPCA	025650	1422*	1590	2117*	2124*	2842*	2929*	2961	3040*	3307	3858*	3876*	3906*	3948*
		4243#												
RPCS	025640	1351*	1424*	1428	1465	1470*	1471	1515*	1519	1535	1560	1566	1604*	1605
		1614*	1615	1626	1628	1631	1661*	1662	1696	1698	1708*	1709	1722	1724
		1736*	1737	1789	1791	1797	1799	1822	1832	1869	1871	1878	1880	1887
		1889	1892	1935	1937	1953	1959	1968	1970	1990*	1999*	2000	2008	2022*
		2033	2035	2042	2044	2048	2067	2082	2086	2103	2105	2115*	2119*	2120
		2125*	2126	2129*	2130	2132	2139*	2140	2160*	2161	2177*	2224	2226	2237
		2239	2242	2277	2279	2291	2293	2296	2327	2333	2337	2376	2378	2382
		2389	2391	2395	2405	2416	2418	2422	2426	2434	2436	2440	2450	2591
		2600*	2601	2752	2762*	2763	2843*	2852	2858	2860	2869	2876*	2883*	2936*
		2944	2953	2998*	2999	3042*	3043*	3045	3048	3059*	3060	3172*	3303	3520*
		3521	3859*	3860	3865	3870*	3871	3877*	3878	3883	3888*	3889	3915*	3935*
		3936	3940	3952*	3953	3955	4239#							
RPCS1	025642	1352*	3934*	4240#	4455*									
RPDA	025652	1571	2118*	2928*	2962	3311	3911*	3949*	4244#					
RPDA1	025654	1578	4245#											
RPDS	025660	1353	1421*	1434	1440	1455	1468	1473	1525	1602	1607	1617	1659	1664
		1706	1711	1734	1739	2078	2122	2137	2142	2178	2603	2765	2854	2884
		2993	2996	3001	3041*	3050	3062	3065	3173	3295	3862	3864*	3868	3873
		3875*	3880	3882*	3886	3891	3893*	3905*	3916	3938	3943	4247#	4456	4461
		4466												
RPER	025656	1824	1955	2002	2090	2329	2964	3299	4246#					
RPWC	025644	2128*	2931*	2932*	3950*	4241#								
RRANEX	013456	2659#	2790											
RW =	000006	861#	1187	1188	1189	2355	2356	2357	2358	3566	3633	3636	3639	3642
RWRCK	006274	1774#	2182											
SALDR	021562	3612	3617#											
SANHT	005412	1640#	1644											
SAVE	025712	3396*	3399*	3426	3430*	3440	3526*	3527	3528	3531*	3536*	3541*	3542	3544*
		3546*	3547	3548*	3550	4264#								

SAVES	001036	936#	1052	1114										
SCOPE =	104000	868#	1476	1539	1610	1667	1714	1742	1837	1914	1983	2013	2062	2094
		2109	2145	2170	2257	2283	2299	2313	2341	2409	2455	2523	2537	2618
		2704	2713	2780	2864	3947	3960							
SCOPES	001004	926#	3183											
SCYL	025666	1292*	3079	4254#										
SEABUF	004710	1544#	1612											
SEEK	015666	2514	2552	2695	2716	3015#								
SEEK1	025752	1197*	3020*	3021*	3022	3024	3040	3057	4280#					
SEEND	016164	3016	3069#											
SEKDON	016166	3038	3070#											
SHED	025670	1300*	3081	4255#										
SKTST	010172	2115#	2144											
SPECME	024532	1216	4105#											
SRO =	177572	844#	1183	1190*	2168*	2322*	2348	2360*	2466*	3519*	3562*	3619*	3629*	
SSEC	025672	1308*	3080	4256#										
START	002336	914	1170#											
STATUS	025636	1415*	2500*	3037	3039*	3067*	3186*	4238#						
STKPTR=	000500	822#	1171	2467	2838	2919	3554							
SUBSIZ	027530	1177	4448#											
SUCA	025662	1459	1462	2886	2889	3315	4248#							
SWITCH	025702	3400*	3407	3410*	3412*	3415	3419*	3421*	4260#					
SWR =	177570	818#	926	959	962	966	976	992	1010	1013	1179	1204	1209	1379
		1381	1487	1670	1754	2180	2460	2494*	2517	2531	2556	2558	2644	2698
		2707	2745	2750	2788	2950	3015	3444	3502	3907	3918			
SWRDCT	025710	1217	1271	1275*	1276	2498	3127	3553*	3610*	3614*	3615*	3731	4263#	
SWSUR	016352	3107	3110#											
T =	000020	794#												
TBITVE=	000014	806#												
TESTNO	025730	1409*	1412	1508*	1511	1681*	1684	1768*	1771	2194*	2197	2494*	2497	2653*
		2656	3317	3319	4271#									
TIMO	024475	3801	4097#											
TKB =	177562	815#	1095											
TKS =	177560	814#												
TKSR	003176	1263	1278#											
TPB =	177566	817#	1005*	1095*										
TPS =	177564	816#	996	1006	1096									
TRAPVE=	000034	810#												
TSTTBL	004016	1385	1390#											
TTY	001562	973*	1027*	1029*	1031*	1050#	1217*	1377*	1412*	1511*	1684*	1771*	1804*	1829*
		2197*	2487*	2614*	2656*	2776*	2813*	2903*	2957*	2980*	2986*	2991*	3057*	3295*
		3299*	3303*	3307*	3311*	3315*	3324*	3337*	3341*	3345*	3349*	3353*	3358*	3362*
		3479*	3485*	3490*	3496*	3500*	3647*	4453*						
UP =	000000	862#	1187	1188	1189	2355	2356	2357	2358	3566	3633	3636	3639	3642
V =	000002	791#												
VECTOR	025634	1414*	2499*	3036	3038*	3068*	3185*	4237#						
WCCON	003106	1264#	1270	1274										
WDERR	020346	3427	3438#											
WORK	025734	1271*	1272*	1273	1570*	1583*	1592	2171*	2173*	2174*	2175*	2176*	2177	2669*
		2677*	2679*	2680	2682	2802*	2803*	2804	2877*	2879*	2880*	2881*	2882*	2883
		3095*	3096*	3097	3104*	3105*	3106	3130*	3132*	3136	3138*	3140*	3141	3143*
		3145*	3166*	3167*	3168*	3170*	3171*	3172	3194*	3195	3197*	3199*	3200*	3201
		3230*	3240*	3244*	3247*	3276*	3280*	3452*	3453*	3454*	3458*	3494	3908*	3909*
		3910*	3911	3912*	3913*	3914*	3915	4273#						
WORK1	025736	2678*	2683*	2684	2686*	2687	2689*	2690	3094*	3134*	3459*	3464	3466*	3469*
		3469	3471*	3473*	3482	3487	4274#							

[illegible]

RP11C RELIABILITY TEST MACY11 27(1006) 29-APR-77 14:49 PAGE 95
 DZRPBE.P11 29-APR-77 14:41 CROSS REFERENCE TABLE -- MACRO NAMES

BUSTST	3734#	3744	3747	3750	3753	3756	3759	3762	3765	3769	3772	3775	3778	3781	3784
	3787	3790													
DUMP	904#	973	1804	1829	2903	3295	3299	3303	3307	3311	3315	3358	3362		
DUMPF	902#														
LDPDR	864#	1187	1188	1189	2355	2356	2357	2358	3566	3633	3636	3639	3642		
PACK	910#	1235	1251	1268	1289	1297	1305	1314	3833	3841	3849	3931			
PRINT	899#	968	1045	1091	1101	1364	1410	1509	1656	1682	1769	1802	1827	2195	2485
	2611	2654	2773	2811	2865	2901	2905	2955	2977	2984	2989	3054	3291	3293	3297
	3301	3305	3309	3313	3321	3331	3335	3339	3343	3347	3351	3355	3360	3476	3483
	3488	3492	3498	3800											
PRINTF	900#	964	1173	1214	1219	1225	1231	1247	1258	1264	1278	1285	1293	1301	1310
	1324	1330	1337	1371	3643	3816	3822	3829	3837	3845	3901	3927			
RAND	907#	2675	3417												
READ	908#	1221	1227	1234	1250	1261	1267	1281	1288	1296	1304	1313	1326	1333	1340
	3818	3825	3832	3840	3848	3930									
REST	896#	1076	1151												
SAVE	895#	1051	1113												
SDUMP	905#	1412	1511	1684	1771	2197	2487	2614	2656	2776	2813	2957	2980	2996	2991
	3057	3324	3337	3341	3345	3349	3353	3479	3485	3490	3496	3500			
SDUMPF	903#	1217	1377	3647											
.DUMP	901#	1009													
.ERROR	897#	956													
.PACK	909#	1110													
.PRINT	898#	990													
.RAND	906#	1051													
.READ	907#	1081													
.REST	895#	945													
.SAVE	895#	934													
.SCOPE	895#	918													

. ABS. 027670 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DSKZ:DZRPBE.BIN,DSKZ:DZRPBE.SEG/CRF/SOL/NL:TOC=DSKM:DZRPBE.P11
 RUN-TIME: 4 9 1 SECONDS
 RUN-TIME RATIO: 166/15=10.4
 CORE USED: 15K (29 PAGES)