

# RP11-C/RP03

RELIABILITY DIAGNOSTIC  
MD-11-DZRPB-D

EP DZRPB-D DL A

OCT 1976

COPYRIGHT ©1976

**digital**

FICHE 1 OF 1

Made in U.S.A.

The microfiche card contains 100 frames of data, arranged in a 10x10 grid. Each frame displays a different set of diagnostic information for an MD-11 aircraft. The data is presented in various formats, including tables, lists, and diagrams. The frames are separated by a grid of small squares. The data in the frames includes various parameters and values, likely related to the aircraft's reliability and performance. The frames are arranged in a grid that is 10 frames wide and 10 frames high. The data in the frames is presented in a clear, organized manner, making it easy to read and interpret. The microfiche card is a common format for storing and distributing large amounts of data, and it is widely used in various industries, including aviation. The data in the frames is likely to be used for troubleshooting and maintenance of the aircraft, and it is an essential part of the aircraft's diagnostic system. The microfiche card is a valuable tool for pilots and maintenance personnel, and it is an important part of the aircraft's documentation. The data in the frames is presented in a clear, organized manner, making it easy to read and interpret. The microfiche card is a common format for storing and distributing large amounts of data, and it is widely used in various industries, including aviation. The data in the frames is likely to be used for troubleshooting and maintenance of the aircraft, and it is an essential part of the aircraft's diagnostic system. The microfiche card is a valuable tool for pilots and maintenance personnel, and it is an important part of the aircraft's documentation.



801

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRPB-C-D

PRODUCT NAME: RPI11 RELIABILITY DIAGNOSTIC

DATE CREATED: DECEMBER, 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOE STUBBLEBINE, WARREN MONCSKO

COPYRIGHT (C) 1972, 1973  
DIGITAL EQUIPMENT CORPORATION  
MAYNARD, MASS.

RELIABILITY TEST MAINDEC 27 (732) 16-SEP-76 16:19 PAGE 1  
DZRPB-P11



1.0 ABSTRACT

THIS PROGRAM TESTS BOTH THE ADDRESSING CAPABILITY AND THE DATA RELIABILITY OF THE RP11C AND THE RPO3. THE PROGRAM CONSIST OF SEVEN TESTS ANY ONE OF WHICH IS SELECTABLE BY THE OPERATOR. A CONVERSATION MODE EXISTS WHICH ALLOWS THE OPERATOR TO DEFINE TEST PARAMETERS.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD FAMILY PROCESSOR  
RP11C DISK PACK CONTROLLER WITH UP TO EIGHT RPO3 DRIVES  
ASR33 OR EQUIVELANT.

2.2 STORAGE

BK OF STORAGE IS REQUIRED TO RUN THIS PROGRAM.

2.3 PRELIMINARY PROGRAMS

DZRPA RP11C DISKLESS DIAGNOSTIC

3.0 LOADING PROEDURE

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).

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
A  
B  
C  
D  
E  
F  
G  
H  
I  
J  
K  
L  
M  
N  
O  
P  
Q  
R  
S  
T  
U  
V  
W  
X  
Y  
Z  
[  
\  
]  
^  
\_  
`  
a  
b  
c  
d  
e  
f  
g  
h  
i  
j  
k  
l  
m  
n  
o  
p  
q  
r  
s  
t  
u  
v  
w  
x  
y  
z  
{  
|  
}  
~  
?  
@  
#  
\$  
%  
&  
\*  
+  
,  
-  
.  
:  
;<br/>=>  
?  
@  
#  
\$  
%  
&  
\*  
+  
,  
-  
.  
:  
;<br/>=>

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

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.

193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248

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.

?

GO1

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:19 PAGE 6  
DZRPB.P11

249  
250

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

251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306

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
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



307

SETS THE ERROR FLAG USED BY THE SCOPE ROUTINE.

308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363

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

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 |

K01

364  
365

AND RANDOM TEST.  
INDICATES THE NUMBER



366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421

OF READS OF THAT  
BLOCK. 0 = FIRST  
READ. IF LOOPING ON  
ERROR THE FIRST READ  
COUNTER ONLY IS  
VALID.  
PC OF FAILURE

2. PC=  
COMPARE ERROR  
EXPECTED=  
RECEIVED=  
CYLINDER=  
HEAD=  
SECTOR=  
WORD COUNT INTO SECTOR=  
READ NO.

DATA EXPECTED  
DATA RECEIVED  
CYLINDER ADDRESS OF  
THE ERROR  
HEAD ADDRESS OF THE  
ERROR  
SECTOR ADDRESS OF  
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

MO1

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:19 PAGE 12  
DZRPB.P11

422  
423

LOOKING AT THE DISPLAY. IT WILL CONTAIN THE NUMBER  
OF THE PATTERN CURRENTLY IN USE.

424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477

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	175716	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





534  
535

SECTOR WITH ZEROS WHEN A PARTIAL SECTOR IS WRITTEN.  
IN THE WRITE CHECK PORTION A FLOATING ONE AND A



536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591

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 CONSIST 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 2BK) 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 UNTILL 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



E02

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:19 PAGE 17  
DZRPB.P11

592  
593

ERROR THE READ RETRY NUMBER IS TYPED OUT ALONG WITH  
THE ERROR INFORMATION. THIS WAY IT CAN BE

594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649

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
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

G02

650

FAILURE.

Handwritten mark

Handwritten mark



651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706

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.  
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)?

707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762

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)

*eg 1/2*

J02

763  
764

WRITE CHECK? (Y OR N)  
READ? (Y OR N)

765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
  
000001  
000002  
000004  
000010  
000020  
0000340  
0000300  
0000240  
0000200  
0000140  
0000100  
000040  
  
000004  
000010  
000014  
000020  
000024  
000030  
000034  
  
177776  
177560  
177562  
177564  
177566

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

.LIST ME  
.NLIST MC,MD,CND  
.ABS  
.TITLE FRONT END  
;COPYRIGHT 1972,1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.  
;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS  
;GENERAL REGISTER ASSIGNMENTS

RO=%0  
R1=%1  
R2=%2  
R3=%3  
R4=%4  
R5=%5  
SP=%6  
PC=%7

;STATUS REGISTER (PSW) BIT ASSIGNMENTS

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

;VECTOR ADDRESSES

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

;REGISTER ADDRESSES

PSW=177776 ;PROCESSOR STATUS REGISTER  
TKS=177560 ;KEYBOARD CSR  
TKB=177562 ;ADDR OF KEYBOARD BUFFER  
TPS=177564 ;TELEPRINTER CSR  
TPB=177566 ;TELEPRINTER BUFFER

FRONT END  
DZRPB.P11

MACY11 27(732) 16-SEP-76 16:19 PAGE 24

```

821      177570      SWR=177570      ;CONSOLE SWITCH REGISTER
822      177570      DISPLAY=177570  ;CONSOLE DISPLAY REGISTER
823
824      ;INITIAL STACK POINTER
825      000500      STKPTR=500      ;PROGRAM STACK POINTER
826
827      ;BIT ASSIGNMENTS
828      100000      B15=100000
829      040000      B14=40000
830      020000      B13=20000
831      010000      B12=10000
832      004000      B11=4000
833      002000      B10=2000
834      001000      B9=1000
835      000400      B8=400
836      000200      B7=200
837      000100      B6=100
838      000040      B5=40
839      000020      B4=20
840      000010      B3=10
841      000004      B2=4
842      000002      B1=2
843      000001      B0=1
844
845      ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS
846
847      177572      SRO=177572
848      172340      KIPAR0=172340
849      172342      KIPAR1=172342
850      172344      KIPAR2=172344
851      172346      KIPAR3=172346
852      172350      KIPAR4=172350
853      172352      KIPAR5=172352
854      172354      KIPAR6=172354
855      172356      KIPAR7=172356
856      172300      KIPDR0=172300
857      172302      KIPDR1=172302
858      172304      KIPDR2=172304
859      172306      KIPDR3=172306
860      172310      KIPDR4=172310
861      172312      KIPDR5=172312
862      172314      KIPDR6=172314
863      172316      KIPDR7=172316
864      000006      RW=6
865      000000      UP=00
866
867
868      ;INSTRUCTION EQUATES
869      104400      -HLT=TRAP      ;HLT IS A TRAP TO THE ERROR ROUTINE
870
871      104000      SCOPE=EMT      ;SCOPE IS AN EMT TRAP
872
873
874      ;INDEX OF MACROS
875      .SCOPE
876      .SAVE

```

877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932

.REST  
.ERROR  
.PRINT  
.DUMP  
.RAND  
.READ  
.PACK  
  
INDEX OF CALLS  
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.  
SCOPES: BIT #B14,2#SWR ;LOOP ON ERROR?  
BEQ 2\$ ;BRANCH IF NO  
TST ERRFLG ;IS THERE AN ERROR?  
BNE 1\$ ;BRANCH IF YES

000200 000200 002336  
000200 012707 002336  
001000 001000  
001002 000000  
  
001004 032737 040000 177570  
001012 001403  
001014 005767 000220  
001020 001003



```

933 001022 005067 000212 2$: CLR ERRFLG ;RESET ERROR CONDITION
934 001026 000002 RTI ;EXIT
935 001030 016716 177746 1$: MOV LAD,(SP) ;MODIFY RETURN ADDRESS
936 001034 000002 RTI ;EXIT
937 ;ROUTINE TO SAVE REGISTERS ON THE STACK.
938 ;CALLED BY SAVE MACRO
939 001036 012667 000020 SAVES: MOV (SP)+,1$ ;SAVE RETURN PC
940 001042 010546 MOV R5,-(SP)
941 001044 010446 MOV R4,-(SP)
942 001046 010346 MOV R3,-(SP)
943 001050 010246 MOV R2,-(SP)
944 001052 010146 MOV R1,-(SP)
945 001054 010046 MOV R0,-(SP)
946 001056 016707 000000 MOV 1$,PC ;RETURN
947 001062 000000 1$: 0 ;CONTAINS RETURN ADDRESS
948 ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
949 ;CALLED BY REST MACRO
950 001064 012667 000020 RESTS: MOV (SP)+,1$ ;SAVE RETURN PC
951 001070 012600 MOV (SP)+,R0
952 001072 012601 MOV (SP)+,R1
953 001074 012602 MOV (SP)+,R2
954 001076 012603 MOV (SP)+,R3
955 001100 012604 MOV (SP)+,R4
956 001102 012605 MOV (SP)+,R5
957 001104 016707 000000 MOV 1$,PC ;RETURN
958 001110 000000 1$: 0 ;CONTAINS RETURN ADDR
959 ;ERROR SERVICE ROUTINE CALLED BY HLT
960 ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
961 ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
962 001112 005737 177570 ERROR: TST @#SWR ;HALT ON ERROR?
963 001116 100001 BPL 3$ ;BRANCH IF NO
964 001120 000000 HALT
965 001122 032737 004000 177570 3$: BIT #B11,@#SWR ;RING THE BELL?
966 001130 001403 BEQ 1$ ;BRANCH IF NO
967 001132 004567 000144 JSR R5,PRNTF$ ;FORCE PRINT THE MESSAGE
968 001136 001250 BELL
969 001140 032737 020000 177570 1$: BIT #B13,@#SWR ;SKIP TYPEOUT?
970 001146 001022 BNE 2$ ;BRANCH IF YES
971 001150 004567 000110 JSR R5,PRINT$ ;PRINT MESSAGE
972 001154 001252 ERRPC
973 001156 011667 000062 MOV (6),HLTADS ;GET ERROR PC+2
974 001162 162767 000002 000054 SUB #2,HLTADS ;MODIFY
975 001170 117767 000050 000044 MOVB @HLTADS,HLTCTS ;SAVE HLT ARGUMENT
976 001176 016767 000042 000356 MOV HLTADS,TTY
977 001204 004767 000134 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
978 001210 004767 016102 JSR PC,MSG ;GO TO USER ERROR ROUTINE
979 001214 005737 177570 2$: TST @#SWR ;HALT ON ERROR?
980 001220 100001 BPL 4$ ;BRANCH IF NO
981 001222 000000 HALT
982 001224 052767 100000 000006 4$: BIS #B15,ERRFLG ;SET ERROR FLAG
983 001232 005267 000010 INC ERRORS ;UPDATE ERROR COUNTER
984 001236 000002 RTI
985 001240 000000 ERRFLG: 0
986 001242 000000 HLTCTS: 0
987 001244 000000 HLTADS: 0 ;PC OF ERROR
988 001246 000000 ERRORS: 0 ;ERROR COUNT

```

```

999 001250 000007
990 001252 005015 005015 041520 BELL: .ASCIZ (<7>
991 001260 020075 000 ERRPC: .ASCIZ (<15><12><15><12>'PC= '
992 001264 001264
993 .EVEN
994 :THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
995 001264 032737 020000 177570 PRINTS: BIT #B13,2#SWR ;INHIBIT TYPEOUTS?
996 001272 001403 BEQ PRNTFS ;BRANCH IF NO
997 001274 062705 000002 ADD #2,R5 ;UPDATE RETURN ADDR
998 001300 000205 RTS R5
999 001302 105737 177564 PRNTFS: TSTB 2#TPS ;WAIT FOR PRINTER TO FINISH
1000 001306 100375 BPL -4
1001 001310 010546 MOV R5,-(SP)
1002 001312 062716 000002 ADD #2,(SP) ;ADJUST RETURN PC
1003 001316 011505 MOV (R5),R5 ;GET MESSAGE ADDR
1004 001320 105715 1S: TSTB (R5) ;CHECK FOR TERMINATOR
1005 001322 001002 BNE 2S
1006 001324 012605 MOV (SP)+,R5 ;GET RETURN ADDR
1007 001326 000205 RTS R5 ;RETURN
1008 001330 112537 177566 2S: MOVB (R5)+,2#TPB ;PRINT CHARACTER
1009 001334 105737 177564 TSTB 2#TPS ;WAIT TILL DONE
1010 001340 100375 BPL -4
1011 001342 000766 BR 1S
1012 :THIS ROUTINE TYPES A LOCATION IN OCTAL
1013 001344 032737 020000 177570 PRINTR: BIT #B13,2#SWR ;INHIBIT TYPEOUT?
1014 001352 001406 BEQ PRINTA ;BRANCH IF NO
1015 001354 000207 RTS PC
1016 001356 032737 020000 177570 PRINTS: BIT #B13,2#SWR ;INHIBIT TYPEOUT?
1017 001364 001406 BEQ PRINTB ;BRANCH IF NO
1018 001366 000207 RTS PC
1019 001370 112767 000001 000140 PRINTA: MOVB #1,.PR ;SET ZERO FILL SWITCH
1020 001376 000402 BR .+6 ;SKIP
1021 001400 005067 000132 PRINTB: CLR .PR ;SUPPRESS LEADING ZEROS
1022 001404 112767 177772 000125 MOVB #-6,.PR+1 ;SET COUNT
1023 001412 010446 .PTIT: MOV R4,-(SP) ;SAVE R4
1024 001414 012704 001540 MOV #.PR+2,R4 ;SET POINTER TO FIRST CHARACTER
1025 001420 105014 CLRB (R4) ;CLEAR FIRST BYTE
1026 001422 000413 BR .PRF ;ROTATE FIRST BIT
1027 001424 105014 .PRL: CLRB (R4) ;CLEAR BYTE OF CHAR
1028 001426 032767 000100 000102 BIT #100,.PR ;BIT TYPING MODE
1029 001434 001006 BNE .PRF ;YES SKIP 2 ROTATES
1030 001436 006167 000120 ROL TTY ;ROTATE BIT INTO C
1031 001442 106114 ROLB (4) ;PACK IT
1032 001444 006167 000112 ROL TTY
1033 001450 106114 ROLB (4)
1034 001452 006167 000104 .PRF: ROL TTY
1035 001456 106114 ROLB (4)
1036 001460 105714 TSTB (4) ;IS IT ZERO
1037 001462 001402 BEQ .+6 ;SKIP INC
1038 001464 105267 000046 INCB .PR ;SET FILL SWITCH
1039 001470 105767 000042 TSTB .PR ;CHECK FILL SWITCH
1040 001474 001402 BEQ .+6 ;SKIP BITSET
1041 001476 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCIZ CHAR
1042 001502 105267 000031 INCB .PR+1 ;INC COUNT
1043 001506 001346 BNE .PRL ;REPEAT
1044 001510 022704 001540 CMP #.PR+2,R4 ;EMPTY BUFFER

```

1045	001514	001002		BNE	+.6	;SKIP IF NOT
1046	001516	112724	000060	MOVB	#'0,(4)+	;LOAD ONE ZERO
1047	001522	105014		CLRB	(4)	;NULL TERMINATOR
1048	001524	004567	177534	JSR	RS,PRINTS	;PRINT MESSAGE
1049	001530	001540		.PR+2		
1050	001532	012604		MOV	(SP)+,R4	;RESTORE R4
1051	001534	000207		RTS	PC	
1052	001536	000012		.BLKW	12	
1053	001562	000000				
1054	001564					
1055	001564	004767	177246	JSR	PC,SAVE\$	;SAVE THE REGISTERS
1056	001570	016700	000106	MOV	LONUM,R0	;SET R0 WITH LOW
1057	001574	016701	000100	MOV	HINUM,R1	;SET R1 WITH HIGH
1058	001600	012703	177771	MOV	#-7,R3	;SET SHIFT COUNT
1059	001604	005002		CLR	R2	
1060	001606	006300		ASL	R0	;SHIFT R0 LEFT AND
1061	001610	006101		ROL	R1	;ROTATE CARRY INTO R1 AND
1062	001612	006102		RCL	R2	;ROTATE CARRY INTO R2
1063	001614	005203		INC	R3	;CHECK FOR DONE
1064	001616	001373		BNE	1\$	
1065	001620	066702	000056	ADD	LONUM,R2	;ADD # TO MAKE X 129
1066	001624	005501		ADC	R1	;PROPOGATE CARRY
1067	001626	066701	000046	ADD	HINUM,R1	;ADD # TO MAKE X 129
1068	001632	005502		ADC	R2	;PROPOGATE CARRY
1069	001634	062700	001057	ADD	#1057,R0	
1070	001640	005501		ADC	R1	;PROPOGATE CARRY
1071	001642	005502		ADC	R2	;PROPOGATE CARRY
1072	001644	062701	047401	ADD	#47401,R1	
1073	001650	005502		ADC	R2	
1074	001652	062702	000006	ADD	#6,R2	
1075	001656	060200		ADD	R2,R0	
1076	001660	005501		ADC	R1	
1077	001662	010067	000014	MOV	R0,LONUM	
1078	001666	010167	000006	MOV	R1,HINUM	
1079	001672	004767	177166	JSR	PC,REST\$	;RESTORE THE REGISTERS
1080	001676	000207		RTS	PC	
1081						
1082	001700	000000		HINUM:	0	
1083	001702	000000		LONUM:	0	
1084	001704	010346		READS:	MOV	R3,-(6)
1085	001706	012703	002014	1\$:	MOV	#INPUT\$,R3
1086	001712	022703	002034	2\$:	CMP	#INPUT\$+20,R3
1087	001716	001412			BEO	4\$
1088	001720	105737	177560		TSTB	@#177560
1089	001724	100375			BPL	.-4
1090	001726	113713	177562		MOVB	@#177562,(3)
1091	001732	142713	000200		BICB	#200,(3)
1092	001736	122713	000177		CMPB	#177,(3)
1093	001742	001004			BNE	3\$
1094	001744			4\$:		
1095	001744	004567	177314		JSR	RS,PRINTS
1096	001750	002054			READM\$	
1097	001752	000755			BR	1\$
1098	001754	013737	177562	177566	3\$:	MOV
1099	001762	105737	177564		TSTB	@#TPB
1100	001766	100375			BPL	.-4

```

1101 001770 122723 000015      CMPB    #15,(3)+      ;CHECK FOR RETURN
1102 001774 001346              BNE     25           ;LOOP IF NOT RETURN
1103 001776 105063 177777      CLR     -1(3)       ;REMOVE THE RETURN
1104 002002 004567 177256      JSR     RS,PRINTS   ;PRINT MESSAGE
1105 002006 002060              READLS
1106 002010 012603              MOV     (6)+,R3     ;RESTORE R3
1107 002012 000207              RTS     PC          ;RETURN
1108
1109 002014 000020      - INPUTS: .BLKW    20
1110 002054 006477 000012      READMS: .ASCIZ   '?',<15><12>
1111 002060 000012      READLS: .ASCIZ   <12>
1112
1113      ;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
1114      ;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
1115
1116 002062      PACKS:
1117 002062 004767 176750      JSR     PC,SAVES   ;SAVE THE REGISTERS
1118 002066 005067 000242      CLR     NUMS
1119 002072 005000              CLR     RO
1120 002074 105760 002014      25:    TSTB    INPUTS(RO)
1121 002100 001402              BEQ     15
1122 002102 005200              INC     RO
1123 002104 000773              BR     25
1124 002106 005300              15:    DEC     RO
1125 002110 004767 000166      JSR     PC,PACS    ;GET OCTAL CHAR
1126 002114 016767 000212 000212      MOV     PK$,NUMS  ;PACK FIRST CHAR
1127 002122 004767 000154      JSR     PC,PACS    ;GET OCTAL CHAR
1128 002126 000241              CLC
1129 002130 006167 000176      ROL     PK$
1130 002134 006167 000172      ROL     PK$
1131 002140 006167 000166      ROL     PK$
1132 002144 056767 000162 000162      BIS     PK$,NUMS  ;PACK SECOND CHAR
1133 002152 004767 000124      JSR     PC,PACS    ;GET OCTAL CHAR
1134 002156 000241              CLC
1135 002160 000367 000146      SWAB   PK$
1136 002164 006067 000142      ROR     PK$
1137 002170 006067 000136      ROR     PK$
1138 002174 056767 000132 000132      BIS     PK$,NUMS  ;PACK THIRD CHAR
1139 002202 004767 000074      JSR     PC,PACS    ;GET OCTAL CHAR
1140 002206 000367 000120      SWAB   PK$
1141 002212 000241              CLC
1142 002214 006167 000112      ROL     PK$
1143 002220 056767 000106 000106      BIS     PK$,NUMS  ;PACK FOURTH CHAR
1144 002226 004767 000050      JSR     PC,PACS    ;GET OCTAL CHAR
1145 002232 000367 000074      SWAB   PK$
1146 002236 000241              CLC
1147 002240 006167 000066      ROL     PK$
1148 002244 006167 000062      ROL     PK$
1149 002250 006167 000056      ROL     PK$
1150 002254 006167 000052      ROL     PK$
1151 002260 056767 000046 000046      BIS     PK$,NUMS  ;PACK FIFTH CHAR
1152 002266 000402              BR
1153 002270 062706 000002      PKEX$: ADD     #2,SP ;MODIFY STACK
1154 002274              PKEX1$:
1155 002274 004767 176564      JSR     PC,RESTS  ;RESTORE THE REGISTERS
1156 002300 000207              RTS     PC          ;EXIT

```



.TITLE RP11C RELIABILITY TEST

1169									
1170									
1171									
1172									
1173	002336	000005				START:	RESET		;CLEAR THE WORLD
1174	002340	012706	000500				MOV	#STKPTR,SP	;SETUP STACK
1175	002344	004767	014272				JSR	PC,INIT	;INITIALIZE VECTORS
1176	002350	004567	176726				JSR	RS,PRNTRF\$	;FORCE PRINT THE MESSAGE
1177	002354	025726					HEADER		
1178	002356	005067	023250				CLR	FLAG	;CLEAR PROGRAM FLAG
1179	002362	032737	000100	177570			BIT	#86,2#SWR	;WANT TO USE MEMORY MANAGEMENT?
1180	002370	001034					BNE	2\$	;BR IF NO
1181	002372	012737	002462	000004			MOV	2\$ 2#ERRVEC	;SETUP TRAP TEST FOR MEMORY MANAGEMENT
1182	002400	012737	000340	000006			MOV	#340,2#ERRVEC+2	
1183	002406	005737	177572				TST	2#SR0	;MEMORY MANAGEMENT?
1184	002412	005037	172340				CLR	2#KIPAR0	;YES! SET UP TO USE MEMORY MANAGEMENT
1185	002416	012737	000200	172342			MOV	#200,2#KIPAR1	;SECOND 4K PAGE
1186	002424	012737	007600	172356			MOV	#7600,2#KIPAR7	;I/O PAGE
1187	002432	012737	177406	172300			MOV	#400*256.-400+UP+RW,2#KIPDR0	;SET KIPDR0=RW UP 400 BLOCKS
1188	002440	012737	177406	172302			MOV	#400*256.-400+UP+RW,2#KIPDR1	;SET KIPDR1=RW UP 400 BLCKS
1189	002446	012737	177406	172316			MOV	#400*256.-400+UP+RW,2#KIPDR7	;SET KIPDR7=RW UP 400 BLOCKS
1190	002454	005237	177572				INC	2#SR0	;TURN ON MEMORY MANAGEMENT
1191	002460	000403					BR	3\$	
1192	002462	052767	000100	023142	2\$:		BIS	#86,FLAG	;BIT6 SET = NO MEMORY MANAGEMENT
1193	002470	012737	000006	000004	3\$:		MOV	#ERRVEC+2,2#ERRVEC	;RESTORE TRAP CATCHER
1194	002476	005037	000006				CLR	2#ERRVEC+2	
1195	002502	005067	023220				CLR	HEADER	;ASCII TERMINATOR SO ON RESTART THERE IS NO HEADER
1196	002506	004567	016206				JSR	RS,EXTMEN	;SET UP DATA BUFFERS
1197	002512	005067	023202				CLR	SEEK1	;INITIALIZE SEEK RANDOM NUMBER GENERATOR
1198	002516	005067	176256				CLR	ICNT	;CLEAR THE PASS COUNTER
1199	002522	005067	023134				CLR	DSKNOR	;CLEAR UNIT FLAG
1200	002526	005067	023112				CLR	CYLINDER	;CLEAR THE CYLINDER ADDRESS
1201	002532	005067	023110				CLR	DMA	;CLEAR DAR REGISTERS
1202	002536	005067	023110				CLR	PATNU	;CLEAR PATTERN COUNT
1203	002542	005067	023154				CLR	CNTA	;CLEAR READ COUNTER FOR DATA AND RANDOM TEST
1204	002546	032737	000400	177570			BIT	#88,2#SWR	;USE CONVERSATION MODE?
1205	002554	001005					BNE	LCONM	;BRANCH IF YES
1206	002556	052767	070000	023046			BIS	#70000,FLAG	
1207	002564	000167	000756				JMP	ADTST	
1208									;ENTER OPERATOR CONVERSATION MODE
1209	002570	013746	177570			LCONM:	MOV	2#SWR,-(SP)	;PUT SWR ONTO STACK
1210	002574	042716	177774				BIC	#177774,(SP)	;CLEAN OUT UNWANTED SWITCHES
1211	002600	022726	000003				CMP	#3,(SP)+	;TEST 3?
1212	002604	001402					BEQ	1\$	;BR IF YES
1213	002606	000167	017666				JMP	CYLSK	;GO DO ADDRESS CONVERSATION
1214	002612					1\$:			
1215	002612	004567	176464				JSR	RS,PRNTRF\$	;FORCE PRINT THE MESSAGE
1216	002616	024501					SPECMES		;STANDARD WORDS TRANSFERED =
1217	002620	016767	023032	176734			MOV	SWRDCT,TTY	
1218	002626	004767	176546				JSR	PC,PRINTB	;FORCE TYPE LOCATION - SUPPRESS ZEROS
1219	002632	004567	176444				JSR	RS,PRNTRF\$	;FORCE PRINT THE MESSAGE
1220	002636	024530					CON1		;ASK ABOUT DATA TEST ONLY
1221	002640	004767	177040				JSR	PC,READ\$	;INPUT MESSAGE
1222	002644	122767	000131	177142			CMPB	#131,INPUT\$	;TEST FOR YES
1223	002652	001003					BNE	+.10	;BRANCH IF NO
1224	002654	052767	002000	022750			BIS	#810,FLAG	;SET DATA TEST ONLY FLAG



1225	002662	004567	176414		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1226	002666	024563			CON2		; ASK ABOUT MULTI DRIVE MODE
1227	002670	004767	177010		JSR	PC,READS	; INPUT MESSAGE
1228	002674	122767	000131	177112	CMPB	#131,INPUTS	; TEST FOR YES
1229	002702	001040			BNE	DATTES	; BRANCH IF NO
1230	002704	052767	004000	022720	BIS	#B11,FLAG	; SET MULTI UNIT FLAG
1231	002712				DSKDR:		
1232	002712	004567	176364		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1233	002716	024617			CON3		; GET NO. OF UNITS
1234	002720	004767	176760		JSR	PC,READS	; INPUT MESSAGE
1235	002724	004767	177132		JSR	PC,PACKS	; CONVERT INPUT TO A NUMBER
1236	002730	005767	177400		TST	NUMS	; IS IT ZERO
1237	002734	001766			BEQ	DSKDR	
1238	002736	162767	000001	177370	SUB	#1,NUMS	
1239	002744	022767	000010	177362	CMP	#10,NUMS	; IS NO. TOO HIGH
1240	002752	101757			BLOS	DSKDR	
1241	002754	016767	177354	022700	MOV	NUMS,DSKNOR	; SAVE HIGHEST UNIT NO.
1242	002762	042767	177770	022672	BIC	#177770,DSKNOR	
1243	002770	000241			CLC		
1244	002772	006167	022664		ROL	DSKNOR	
1245	002776	006167	022660		ROL	DSKNOR	
1246	003002	000423			BR	ASKWC	
1247	003004				DATTES:		
1248	003004	004567	176272		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1249	003010	024661			CON4		; ASK UNIT NUMBER
1250	003012	004767	176666		JSR	PC,READS	; INPUT MESSAGE
1251	003016	004767	177040		JSR	PC,PACKS	; CONVERT INPUT TO A NUMBER
1252	003022	022767	000010	177304	CMP	#10,NUMS	; IS NO = OR > 10
1253	003030	101765			BLOS	DATTES	; NO
1254	003032	000241			CLC		
1255	003034	006167	177274		ROL	NUMS	
1256	003040	006167	177270		ROL	NUMS	
1257	003044	056767	177264	022560	BIS	NUMS,FLAG	; SAVE UNIT UNDER TEST
1258	003052				ASKWC:		
1259	003052	004567	176224		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1260	003056	024700			CON5		; ASK ABOUT OPTIONAL WORD COUNT
1261	003060	004767	176620		JSR	PC,READS	; INPUT MESSAGE
1262	003064	122767	000131	176722	CMPB	#131,INPUTS	; TEST FOR YES
1263	003072	001034			BNE	TKSR	; ASK ABOUT OPTIONAL DAR
1264	003074				WCCON:		
1265	003074	004567	176202		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1266	003100	024740			CON6		; ASK LENGTH OF WC
1267	003102	004767	176576		JSR	PC,READS	; INPUT MESSAGE
1268	003106	004767	176750		JSR	PC,PACKS	; CONVERT INPUT TO A NUMBER
1269	003112	005767	177216		TST	NUMS	
1270	003116	001766			BEQ	WCCON	
1271	003120	016767	022532	022554	MOV	SWRDCT,WORK	
1272	003126	005267	022550		INC	WORK	
1273	003132	026767	022544	177174	CMP	WORK,NUMS	; IS NO. GREATER THAN AVAILABLE CORE?
1274	003140	101755			BLOS	WCCON	; YES ASK FOR COUNT AGAIN
1275	003142	016767	177166	022506	MOV	NUMS,SWRDCT	; OPERATING WORD COUNT
1276	003150	016767	022502	022464	MOV	SWRDCT,WRDCT	
1277	003156	052767	000002	022446	BIS	#B1,FLAG	; OPERATOR SELECTED WORD COUNT
1278	003164				TKSR:		
1279	003164	004567	176112		JSR	RS,PRNTFS	; FORCE PRINT THE MESSAGE
1280	003170	025011			CON7A		; ASK ABOUT DISK ADDR

1281	003172	004767	176506		JSR	PC, READS	; INPUT MESSAGE
1282	003176	122767	000131	176610	CMPB	#131, INPUTS	; WILL OPERATOR SUPPLY ADDR?
1283	003204	001055			BNE	OPPAT	; BRANCH IF NO
1284	003206	052767	000040	022416	BIS	#B5, FLAG	
1285	003214				OPDAR:		
1286	003214	004567	176062		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1287	003220	025116			CON7C		; GET CYLINDER ADDR
1288	003222	004767	176456		JSR	PC, READS	; INPUT MESSAGE
1289	003226	004767	176630		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
1290	003232	022767	000626	177074	CMP	#626, NUMS	; IS CYLINDER LEGAL
1291	003240	101765			BLOS	OPDAR	
1292	003242	016767	177066	022364	MOV	NUMS, SCYL	; SAVE ADDR
1293	003250				OPDA1:		
1294	003250	004567	176026		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1295	003254	024770			CON7		; GET HEAD ADDR
1296	003256	004767	176422		JSR	PC, READS	; INPUT MESSAGE
1297	003262	004767	176574		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
1298	003266	022767	000024	177040	CMP	#24, NUMS	
1299	003274	101765			BLOS	OPDA1	; BRANCH IF HEAD ADDR TOO HIGH
1300	003276	016767	177032	022332	MOV	NUMS, SHED	; SAVE ADDR
1301	003304				OPDA2:		
1302	003304	004567	175772		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1303	003310	025075			CON7B		; GET SECTOR ADDR
1304	003312	004767	176366		JSR	PC, READS	; INPUT MESSAGE
1305	003316	004767	176540		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
1306	003322	022767	000012	177004	CMP	#12, NUMS	; IS SECTOR ADDR TOO HIGH?
1307	003330	101765			BLOS	OPDA2	
1308	003332	016767	176776	022300	MOV	NUMS, SSEC	; SAVE ADDR
1309							
1310	003340				OPPAT:		
1311	003340	004567	175736		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1312	003344	025136			CON8		; ASK ABOUT DATA PATTERNS
1313	003346	004767	176332		JSR	PC, READS	; INPUT MESSAGE
1314	003352	004767	176504		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
1315	003356	022767	000020	176750	CMP	#20, NUMS	; TEST FOR CORRECT NO
1316	003364	101765			BLOS	OPPAT	; ASK AGAIN
1317	003366	022767	000017	176740	CMP	#17, NUMS	
1318	003374	001411			BEO	OPWRT	; DATA PATTERN UNDER PROGRAM CONTROL
1319	003376	052767	100000	022226	BIS	#B15, FLAG	; SET PROGRAM FLAG
1320	003404	016767	176724	022240	MOV	NUMS, PATNU	; OPERATOR WANTS TO SELECT DATA
1321	003412	000241			CLC		
1322	003414	006167	022232		ROL	PATNU	
1323	003420	042767	070000	022204	OPWRT:	#70000, FLAG	; CLEAR OP MODE BITS IN FLAG
1324	003426	004567	176650		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1325	003432	025164			CON9		; ASK ABOUT WRITE
1326	003434	004767	176244		JSR	PC, READS	; INPUT MESSAGE
1327	003440	122767	000131	176346	CMPB	#131, INPUTS	; TEST FOR YES
1328	003446	001003			BNE	OPRD	; ASK ABOUT WRITE CHECK
1329	003450	052767	040000	022154	BIS	#B14, FLAG	; YES SET FLAG BIT
1330	003456				OPRD:		
1331	003456	004567	176620		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
1332	003462	025234			CON11		; ASK ABOUT READ
1333	003464	004767	176214		JSR	PC, READS	; INPUT MESSAGE
1334	003470	122767	000131	176316	CMPB	#131, INPUTS	; TEST FOR YES ANSWER
1335	003476	001003			BNE	OPWCK	
1336	003500	052767	010000	022124	BIS	#B12, FLAG	; SET FLAG TO READ

1337	003506									
1338	003506	004567	175570							
1339	003512	025205								
1340	003514	004767	176164							
1341	003520	122767	000131	176266						
1342	003526	001003								
1343	003530	052767	020000	022074						
1344	003536	032767	070000	022066	CHKMOD:	BIT	#B13, FLAG			; SET WRITE CHECK FLAG
1345	003544	001725				BEQ	OPWRT			; MAKE SURE SOME OPERATION WAS SELECTED
1346										
1347										
1348	003546	005737	000042		ADTST:	TST	#42			; UNDER MONITOR CONTROL?
1349	003552	001444				BEQ	15			; BRANCH IF NO
1350	003554	005067	022102			CLR	DSKNOR			
1351	003560	012777	000001	022020	3S:	MOV	#1, DRPCS			; CLEAR THE RP11C
1352	003566	116777	022070	022014		MOVB	DSKNOR, DRPCS1			; SELECT THE DRIVE
1353	003574	005777	022026			TST	DRPDS			; IS THE UNIT READY?
1354	003600	100003				BPL	25			; BRANCH IF NO
1355	003602	005267	022054			INC	DSKNOR			; UPDATE UNIT NUMBER
1356	003606	000764				BR	35			
1357	003610	005367	022046		2S:	DEC	DSKNOR			; DSKNOR = NUMBER OF UNITS
1358	003614	000241				CLC				
1359	003616	006167	022040			ROL	DSKNOR			
1360	003622	006167	022034			ROL	DSKNOR			
1361	003626	052767	004000	021776		BIS	#B11, FLAG			; SET MULTI DRIVE FLAG
1362	003634	005767	022022			TST	DSKNOR			; WERE ANY UNITS AVAILABLE?
1363	003640	100011				BPL	15			; BRANCH IF YES
1364	003642	004567	175416			JSR	RS, PRINTS			; PRINT MESSAGE
1365	003646	024207				MES20				
1366	003650	013701	000042			MOV	#42, R1			; ABORT - NO UNITS AVAILABLE
1367	003654	005067	174162			CLR	42			; SET ABORT FLAG
1368	003660	000167	010614			JMP	MEXIT			
1369	003664	032767	004000	021740	1S:	BIT	#B11, FLAG			; ARE WE IN MULTI DRIVE MODE?
1370	003672	001422				BEQ	EXMFLG			; BRANCH IF NO
1371	003674	004567	175402			JSR	RS, PRINTS			; FORCE PRINT THE MESSAGE
1372	003700	024061				MES11				
1373	003702	016767	021724	021764		MOV	FLAG, ACNVX			; TELL OPERATOR THE UNIT UNDER TEST
1374	003710	006067	021760			ROR	ACNVX			
1375	003714	006067	021754			ROR	ACNVX			
1376	003720	042767	177770	021746		BIC	#177770, ACNVX			
1377	003726	016767	021742	175626		MOV	ACNVX, TTY			
1378	003734	004767	175440			JSR	PC, PRINTB			; FORCE TYPE LOCATION - SUPPRESS ZEROS
1379	003740	032737	000010	177570	EXMFLG:	BIT	#B3, #SWR			; RUN SELECTED TEST?
1380	003746	001410				BEQ	15			; BRANCH IF NO
1381	003750	013700	177570			MOV	#SWR, RO			; GET SWITCH SETTINGS
1382	003754	042700	177770			BIC	#177770, RO			
1383	003760	000241				CLC				
1384	003762	006100				ROL	RO			
1385	003764	000170	004004			JMP	DTSTTBL(RO)			; GO TO SELECTED TEST
1386	003770	032767	002000	021634	1S:	BIT	#B10, FLAG			; DATA TEST ONLY?
1387	003776	001412				BEQ	ADT1			; NO
1388	004000	000167	006266			JMP	DATAT			; DO DATA TEST
1389										
1390	004004	004024			TSTTBL:	ADT1				
1391	004006	004502				ADT2				
1392	004010	005564				ADT3				

1393 004012 006230  
1394 004014 010526  
1395 004016 012272  
1396 004020 013366  
1397 004022 014514

WRCK  
MEMTST  
DATAT  
RANEX  
PFTST

1398  
1399  
1400

1401  
1402  
1403

1404  
1405  
1406

1407  
1408

1409 004024 005067 021646  
1410 004030 004567 175230

1411 004034 023740  
1412 004036 016767 021634 175516

1413 004044 004767 175306  
1414 004050 012777 004470 021524  
1415 004056 012777 000340 021520

1416 004064 004567 012500  
1417 004070 005067 021550

1418 004074 005067 021606  
1419 004100 005067 021604

1420 004104 012737 000200 177776  
1421 004112 117777 021510 021506

1422 004120 016777 021520 021470  
1423 004126 005067 000346

1424 004132 052777 020011 021446  
1425 004140 012700 000025

1426 004144 005300  
1427 004146 001376

1428 004150 105777 021432  
1429 004154 100402

1430 004156 104400  
1431 004160 000467

1432 004162 005767 021520  
1433 004166 001406

1434 004170 032777 002000 021430  
1435 004176 001002

1436 004200 104400  
1437 004202 000456

1438 004204 005000  
1439 004206 005200

1440 004210 005777 021412  
1441 004214 100414

1442 004216 005237 025714  
1443 004222 005237 025714

1444 004226 005337 025714  
1445 004232 005337 025714  
1446 004236 005700  
1447 004240 001362  
1448 004242 104400

.SC77L \*\*\*\*\* 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.

```
ADT1: CLR TESTNO
      JSR R5,PRINT$ ;PRINT MESSAGE
      MES6
      MOV TESTNO,TTY
      JSR PC,PRINT$ ;TYPE LOCATION-SUPRESS 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
      MOVB @RPDS,@RPDS ;CLEAR ATTENTION BITS
      MOV CYLINDER,@RPCA ;SET CYLINDER REGISTER
      CLR INTFLG ;CLEAR INTERRUPT FLAG
      BIS #20011,@RPCS ;SEEK AND ENABLE ATTN INTERRUPT
      MOV #25,RO
      1$: DEC RO ;DELAY FOR DONE TO SET
      BNE 1$
      TSTB @RPCS ;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,@RPDS ;DID SEEK UNDERWAY SET?
      BNE 3$ ;BRANCH IF YES
      HLT ;SEEK UNDERWAY DID NOT SET
      BR 8$
      3$: CLR RO
      5$: INC RO ;TIMEOUT UNIT READY
      TST @RPDS ;IS UNIT READY?
      BMI 6$ ;BRANCH IF YES
      INC @CYLA
      INC @CYLA
      DEC @CYLA
      DEC @CYLA
      TST RO ;TIMEOUT?
      BNE 5$ ;BRANCH IF NO
      HLT ;READY DID NOT SET AFTER SEEK
```

1449	004244	000435				BR	8\$	
1450	004246	005767	000226		6\$:	TST	INTFLG	; DID INTERRUPT OCCUR?
1451	004252	001002				BNE	12\$	; BRANCH IF YES
1452	004254	104400				HLT		; INTERRUPT DID NOT OCCUR ON ATTENTION BIT
1453	004256	000430				BR	8\$	
1454	004260	004767	012512		12\$:	JSR	PC, GATTN	; DETERMINE ATTENTION BIT
1455	004264	036777	012536	021334		BIT	ATTN, ARPDS	; IS ATTENTION BIT SET?
1456	004272	001002				BNE	7\$	; BRANCH IF YES
1457	004274	104400				HLT		; ATTENTION BIT DID NOT SET
1458	004276	000420				BR	8\$	
1459	004300	026777	021340	021322	7\$:	CMP	CYLINDER, ASUCA	; IS SUCA CORRECT?
1460	004306	001410				BEQ	11\$	
1461	004310	016767	021330	013440		MOV	CYLINDER, EXP\$	; EXPECTED RESULTS
1462	004316	017767	021306	013434		MOV	ASUCA, RECS	; RECEIVED RESULTS
1463	004324	104401				HLT	+1	; CONTENTS OF SUCA INCORRECT
1464	004326	000404				BR	8\$	
1465	004330	005777	021252		11\$:	TST	ARPCS	; ANY DEVICE ERRORS
1466	004334	100001				BPL	8\$	; BRANCH IF NO
1467	004336	104400				HLT		; DEVICE ERROR AFTER SEEK OPERATION
1468	004340	032777	004000	021260	8\$:	BIT	#B11, ARPDS	; SEEK INCOMPLETE ERROR?
1469	004346	001412				BEQ	4\$	; BRANCH IF NO
1470	004350	112777	000015	021230		MOVB	#15, ARPDS	; ISSUE HOME COMMAND
1471	004356	105777	021224			TSTB	ARPCS	; WAIT FOR DONE
1472	004362	100375				BPL	.-4	
1473	004364	032777	100000	021234	13\$:	BIT	#B15, ARPDS	; WAIT FOR UNIT READY
1474	004372	001774				BEQ	13\$	
1475	004374	012767	004112	174400	4\$:	MOV	#9\$, LAD	; SET UP LOOP
1476	004402	104000				SCOPE		
1477	004404	005767	021300			TST	WORK3	; SEEK CYLINDER ZERO?
1478	004410	100411				BMI	10\$	; BRANCH IF YES
1479	004412	005267	021270			INC	WORK2	; UPDATE CYLINDER
1480	004416	016767	021264	021220		MOV	WORK2, CYLINDER	
1481	004424	052767	100000	021256		BIS	#B15, WORK3	; SET SEEK ZERO FLAG
1482	004432	000627				BR	9\$	
1483	004434	005067	021204		10\$:	CLR	CYLINDER	; CLEAR SEEK ZERO FLAG
1484	004440	005067	021244			CLR	WORK3	; HAS LAST CYLINDER BEEN REACHED?
1485	004444	022767	000625	021234		CMP	#625, WORK2	; BRANCH IF NO
1486	004452	001217				BNE	9\$	; REPEAT TEST
1487	004454	032737	002000	177570		BIT	#B10, ASWR	; NO-GO TO NEXT
1488	004462	001407				BEQ	ADT2	; YES
1489	004464	000167	177360			JMP	RADT1	
1490								
1491								
1492	004470	012767	000001	000002	INTCK:	MOV	#1, INTFLG	; SET INTERRUPT FLAG
1493	004476	000002				RTI		
1494	004500	000000			INTFLG:	0		

1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550

004502 012767 000001 021166  
004510 004567 174550  
004514 023740  
004516 016767 021154 175036  
004524 004767 174626  
004530 005067 015216  
004534 004567 012030  
004540 052777 000015 021040  
004546 012700 000025  
004552 005300  
004554 001376  
004556 105777 021024  
004562 100402  
004564 104400  
004566 000425  
004570 005000  
004572 005200  
004574 005777 021026  
004600 100414  
004602 005237 025714  
004606 005237 025714  
004612 005337 025714  
004616 005337 025714  
004622 005700  
004624 001362  
004626 104400  
004630 000404  
004632 005777 020750  
004636 100001  
004640 104400  
004642 012767 004540 174132  
004650 104000  
004652 012767 005000 020762  
004660 012767 025730 020766  
004666 005067 020754  
004672 005067 020746  
004676 012700 025730  
004702 012701 000400  
004706 016720 020732  
004712 005301  
004714 016720 020726  
004720 005301  
004722 001374

.SBTTL \*\*\*\*\* TEST 1 \*\*\*\*\*

;WRITE 5000 (OCTAL) WORDS IN TEN SECTORS ON EACH TRACK. THE FIRST  
;WORD OF EACH SECTOR IS THE CYLINDER NUMBER AND THE REMAINING WORDS CONTAIN  
;THE HEAD AND SECTOR ADDRESS. THEN EACH SECTOR IS READ BACK TEN AT A TIME AND  
;COMPARED. IF THE FIRST WORD OF A SECTOR DOES NOT COMPARE, THE WRONG  
;CYLINDER WAS PROBABLY SELECTED. A NON COMPARE ON THE FIRST  
;WORD IS INDICATED BY TYPING "CYL" AFTER THE ERRING DATA.  
;IF ANY OTHER WORD FAILS THE WRONG HEAD OR SECTOR WAS  
;SELECTED. THE RIGHT HALF OF THE DATA TYPED EQUALS THE SECTOR  
;AND THE LEFT HALF INDICATES THE HEAD.

ADT2: MOV #1,TESTNO  
JSR R5,PRINT\$ ;PRINT MESSAGE  
MES6  
MOV TESTNO,TTY  
JSR FC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS  
CLR MEX ;CLEAR DRIVE EXTENDED MEMORY BITS  
RADT2: JSR R5,DSKNOS ;SELECT THE DRIVE  
1\$: BIS #15,DRPCS ;SEEK HOME  
MOV #25,RO  
2\$: DEC RO ;GIVE DONE A CHANCE TO SET  
BNE 2\$  
TSTB DRPCS ;IS DONE SET?  
BMI 3\$ ;YES-BRANCH  
HLT ;DONE DID NOT SET AFTER A SEEK HOME  
BR 6\$ ;CHECK FOR LOOPING  
3\$: CLR RO  
5\$: INC RO  
TST DRPDS ;IS UNIT READY?  
BMI 4\$ ;YES BRANCH  
INC @#CYLA  
INC @#CYLA  
DEC @#CYLA  
DEC @#CYLA  
RO  
TST RO ;HAS UNIT TIMED OUT  
BNE 5\$ ;NO-BRANCH  
HLT ;READY DID NOT SET AFTER HOME SEEK  
BR 6\$ ;CHECK FOR LOOPING  
4\$: TST DRPCS ;ANY ERRORS?  
BPL 6\$ ;NO-BRANCH  
HLT ;DRIVE ERRORS AFTER HOME SEEK  
5\$: MOV #1\$,LAD ;SETUP LOOP ADDRESS  
SCOPE  
MOV #5000,WRDCT ;SETUP WORD COUNT FOR 10 SECTORS  
MOV #OUTBUF,BUF ;SETUP OUTPUT BUFFER ADDR  
CLR DMA  
CLR CYLINDER  
SEABUF: MOV #OUTBUF,RO ;GET BUFFER STARTING ADDR  
21\$: MOV #400,R1 ;SECTOR COUNT  
MOV CYLINDER,(RO)+ ;GENERATE PATTERN SO THAT THE  
DEC R1 ;THE FIRST WORD OF EACH SECTOR  
MOV DMA,(RO)+ ;EQUALS THE CYLINDER ADDR AND  
DEC R1 ;THE REMAINDER EQUALS THE HEAD AND  
BNE 1\$ ;SECTOR ADDR



1551	004724	122767	000011	020714		CMPB	#11,DMA	
1552	004732	001403				BEQ	22\$	
1553	004734	005267	020706			INC	DMA	;UPDATE SECTOR COUNT
1554	004740	000760				BR	21\$	
1555	004742	105067	020700		22\$:	CLRB	DMA	
1556	004746	004567	010162		4\$:	JSR	RS,FUNCT	;WRITE TEN SECTORS
1557	004752	000003			.WORD	3		
1558	004754	005000				CLR	RO	
1559	004756	005200			3\$:	INC	RO	
1560	004760	105777	020622			TSTB	DRPCS	;IS DONE SET?
1561	004764	100404				BMI	2\$	;YES BRANCH
1562	004766	005700				TST	RO	;TEST FOR TIMEOUT
1563	004770	001372				BNE	3\$	;BRANCH IF NO
1564	004772	104400				HLT		;DONE DID NOT SET AFTER WRITE
1565	004774	000501				BR	5\$	
1566	004776	005777	020604		2\$:	TST	DRPCS	;ANY DEVICE ERRORS?
1567	005002	100002				BPL	6\$	;BRANCH IF NO
1568	005004	104400				HLT		;RP11C STATUS ERROR AFTER WRITE
1569	005006	000474				BR	5\$	
1570	005010	005067	020666		6\$:	CLR	WORK	;INCREMENT FLAG
1571	005014	017767	020600	012736		MOV	DRPDA,RECS	;GET DISK ADDR
1572	005022	042767	177760	012730		BIC	#177760,RECS	;SAVE SECTOR ADDR
1573	005030	005767	012724			TST	RECS	
1574	005034	001404				BEQ	9\$	;BRANCH IF SECTOR = ZERO
1575	005036	005067	012714			CLR	EXPS	
1576	005042	104401				HLT	+1	;SECTOR ADDR IN RPDA DID NOT UPDATE
1577	005044	000455				BR	5\$	;PROPERLY AFTER A TEN SECTOR WRITE
1578	005046	117767	020550	012704	9\$:	MOVB	DRPDA1,RECS	;GET THE HEAD ADDR
1579	005054	116767	020567	012674		MOVB	DMA+1,EXPS	;SECTOR ADDR OUTPUTTED
1580	005062	122767	000023	012666		CMPB	#23,EXPS	;DID WE OUTPUT HEAD 23?
1581	005070	001005				BNE	7\$	;BRANCH IF NO
1582	005072	005067	012660			CLR	EXPS	;RESET HEAD ADDR
1583	005076	010667	020600			MOV	SP,WORK	;SET INCREMENT FLAG
1584	005102	000402				BR	8\$	
1585	005104	005267	012646		7\$:	INC	EXPS	
1586	005110	126767	012642	012642	8\$:	CMPB	EXPS,RECS	;IS DISK HEAD ADDR CORRECT?
1587	005116	001402				BEQ	12\$	;BRANCH IF YES
1588	005120	104401				HLT	+1	;HEAD ADDR IN RPDA WAS INCORRECT
1589	005122	000426				BR	5\$	;AFTER TEN SECTOR WRITE
1590	005124	017767	020466	012626	12\$:	MOV	DRPCA,RECS	;GET DISK CYLINDER ADDR
1591	005132	016767	020506	012616		MOV	CYLINDER,EXPS	
1592	005140	005767	020536			TST	WORK	;IS INCREMENT FLAG SET?
1593	005144	001410				BEQ	13\$	;BRANCH IF NO
1594	005146	005267	012604			INC	EXPS	
1595	005152	022767	000626	012576		CMP	#526,EXPS	;WAS IT LAST CYLINDER?
1596	005160	001002				BNE	13\$	;BRANCH IF NO
1597	005162	005067	012570			CLR	EXPS	
1598	005166	026767	012564	012564	13\$:	CMP	EXPS,RECS	;IS DISK CYLINDER ADDR CORRECT?
1599	005174	001401				BEQ	5\$	;BRANCH IF YES
1600	005176	104401				HLT	+1	;CYLINDER ADDR IN RPDA IS NOT
1601								;CORRECT AFTER TEN SECTOR WRITE
1602	005200	032777	004000	020420	5\$:	BIT	#811,DRPDS	;SEEK INCOMPLETE ERROR?
1603	005206	001412				BEQ	10\$	;BRANCH IF NO
1604	005210	112777	000015	020370		MOVB	#15,DRPCS	;ISSUE HOME COMMAND
1605	005216	105777	020364			TSTB	DRPCS	;WAIT FOR DONE
1606	005222	100375				BPL	.-4	

1607	005224	032777	100000	020374	11\$:	BIT	#B15,DRPDS	;WAIT FOR UNIT READY
1608	005232	001774				BEQ	11\$	
1609	005234	012767	004746	173540	10\$:	MOV	#45,LAD	;SETUP LOOP ADDR
1610	005242	104000				SCOPE		
1611	005244	004767	010710			JSR	PC,DISBUF	;SETUP NEXT DISK ADDR
1612	005250	000612				BR	SEABUF	;WRITE NEXT SECTOR
1613	005252	012767	005000	020362		MOV	#5000,WROCT	;RESTORE WORD COUNT
1614	005260	052777	000015	020320		BIS	#15,DRPCS	;SEEK HOME
1615	005266	105777	020314			TSTB	DRPCS	
1616	005272	100375				BPL	-4	;WAIT FOR DONE AFTER SEEK HOME
1617	005274	005777	020326			TST	DRPDS	
1618	005300	100375				BPL	-4	;WAIT FOR DRIVE READY AFTER SEEK HOME
1619	005302	012700	025730		RDSECT:	MOV	#OUTBUF,RO	
1620	005306	012701	005000			MOV	#5000,R1	
1621	005312	005020			23\$:	CLR	(R0)+	;CLEAR THE BUFFER
1622	005314	005301				DEC	R1	
1623	005316	001375				BNE	23\$	
1624	005320	004567	007610			JSR	R5,FUNCT	;READ TEN SECTORS
1625	005324	000005			.WORD	5		
1626	005326	105777	020254			TSTB	DRPCS	
1627	005332	100375				BPL	-4	;WAIT FOR DONE AFTER READ
1628	005334	005777	020246			TST	DRPCS	;ANY ERRORS?
1629	005340	100006				BPL	ADHGT	;BRANCH NO ERRORS
1630	005342	104400				HLT		;STATUS ERROR AFTER A READ
1631	005344	032777	040000	020234		BIT	#B14,DRPCS	;WAS IT A DATA ERROR?
1632	005352	001401				BEQ	ADHGT	;IF YES GO COMPARE DATA
1633	005354	000446				BR	ADTER1	
1634	005356	012700	025730		ADHGT:	MOV	#OUTBUF,RO	
1635	005362	012701	000400		ADHGT1:	MOV	#400,R1	
1636	005366	026710	020252			CMP	CYLINDER,(0)	;IS CYLINDER WORD CORRECT?
1637	005372	001017				BNE	ADERC	;BRANCH IF NO
1638	005374	005720				TST	(0)+	
1639	005376	005301				DEC	R1	
1640	005400	026710	020242		SANHT:	CMP	DMA,(0)	;IS HEAD-SECTOR WORD CORRECT?
1641	005404	001016				BNE	ADERR	;BRANCH IF NO
1642	005406	005720				TST	(0)+	
1643	005410	005301				DEC	R1	
1644	005412	001372				BNE	SANHT	
1645	005414	122767	000011	020224		CMPB	#11,DMA	
1646	005422	001423				BEQ	ADTER1	
1647	005424	005267	020216			INC	DMA	
1648	005430	000754				BR	ADHGT1	
1649	005432	016767	020206	012316	ADERC:	MOV	CYLINDER,EXPS	;CORRECT DATA/ADDRESS
1650	005440	000403				BR	ADERC1	
1651	005442	016767	020200	012306	ADERR:	MOV	DMA,EXPS	;CORRECT DATA/ADDRESS
1652	005450	011067	012304		ADERC1:	MOV	(0),RECS	;INCORRECT DATA
1653	005454	104401				HLT	+1	;DATA COMPARE ERROR
1654	005456	022701	000400			CMP	#400,R1	;WAS FIRST WORD INCORRECT?
1655	005462	001003				BNE	ADTER1	;BRANCH IF NO
1656	005464	004567	173574			JSR	R5,PRINT\$	;PRINT MESSAGE
1657	005470	024075				MES12		;WRONG CYLINDER PROBABLY SELECTED
1658	005472	105067	020150		ADTER1:	CLRB	DMA	
1659	005476	032777	004000	020122		BIT	#B11,DRPDS	;SEEK INCOMPLETE ERROR?
1660	005504	001412				BEQ	1\$	;BRANCH IF NO
1661	005506	112777	000015	020072		MOVB	#15,DRPCS	;ISSUE HOME COMMAND
1662	005514	105777	020066			TSTB	DRPCS	;WAIT FOR DONE

1663	005520	100375				BPL	.-4	
1664	005522	032777	100000	020076	25:	BIT	#B15,DRPDS	;WAIT FOR UNIT READY
1665	005530	001774				BEQ	25	
1666	005532	012767	005302	173242	18:	MOV	#RDSECT,LAD	;SETUP LOOP
1667	005540	104000				SCOPE		
1668	005542	004767	010412			JSR	PC,DISBUF	;SETUP NEXT DISK ADDRESS
1669	005546	000655				BR	RDSECT	;CHECK NEXT SECTOR
1670	005550	032737	002000	177570		BIT	#B10,DRSWR	;LOOP ON TEST?
1671	005556	001402				BEQ	ACT3	;BRANCH IF NO
1672	005560	000167	176750			JMP	RADT2	

```

1673 -
1674
1675
1676
1677
1678
1679
1680
1681 005564 012767 000002 020104 ADT3:  MOV  #2,TESTNO
1682 005572 004567 173466          JSR  R5,PRINT$      ;PRINT MESSAGE
1683 005576 023740          MESS
1684 005600 016767 020072 173754  MOV  TESTNO,TTY
1685 005606 004767 173544          JSR  PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
1686 005612 005067 014134          CLR  MEX             ;CLEAR DRIVE EXTENDED MEMORY BITS
1687 005616 004767 011020  RADT3: JSR  PC,INIT        ;INITIALIZE VECTORS
1688 005622 004567 010742          JSR  R5,DSKNOS      ;SELECT UNIT
1689 005626 005067 020014          CLR  DMA
1690 005632 005067 020006          CLR  CYLINDER
1691 005636 012767 000001 017776  MOV  #1,WRDCT        ;SET UP WORD COUNT
1692 005644 012767 025730 020002  MOV  #OUTBUF,BUF     ;SETUP BUFFER ADDR
1693 005652 016737 017766 025730  WRCYL: MOV  CYLINDER,#OUTBUF ;INSERT PATTERN
1694 005660 004567 007250          JSR  R5,FUNCT       ;WRITE PATTERN ON FIRST SECTOR
1695 005664 000003  .WORD 3              ;OF CYLINDER
1696 005666 105777 017714          TSTB 2RPCS          ;WAIT FOR DONE
1697 005672 100375          BPL  -4
1698 005674 005777 017706          TST  2RPCS          ;AND ERRORS?
1699 005700 100002          BPL  1$             ;BRANCH IF NO
1700 005702 104400          HLT
1701 005704 000407          BR   2$
1702 005706 022767 000625 017730  1$:  CMP  #625,CYLINDER  ;ALL CYLINDERS WRITTEN?
1703 005714 001403          BEQ  2$             ;BRANCH IF YES
1704 005716 005267 017722          INC  CYLINDER
1705 005722 000753          BR   WRCYL
1706 005724 032777 004000 017674  2$:  BIT  #B11,2RPCS     ;SEEK INCOMPLETE ERROR?
1707 005732 001412          BEQ  3$             ;BRANCH IF NO
1708 005734 112777 000015 017644  MOVB #15,2RPCS      ;ISSUE HOME COMMAND
1709 005742 105777 017640          TSTB 2RPCS          ;WAIT FOR DONE
1710 005746 100375          BPL  -4
1711 005750 032777 100000 017650  4$:  BIT  #B15,2RPCS     ;WAIT FOR UNIT READY
1712 005756 001774          BEQ  4$
1713 005760 012767 005652 173014  3$:  MOV  #WRCYL,LAD     ;SETUP UP LOOP
1714 005766 104000          SCOPE
1715 005770 005067 017650          CLR  CYLINDER
1716 005774 012767 000624 017704  MOV  #624,WORK2
1717 006002 005067 017702          CLR  WORK3
1718 006006 005067 017700          CLR  WORK4
1719 006012 016767 017652 017634  ADT32: MOV  INBUF,BUF     ;INC - DEC FLAG
1720 006020 004567 007110          JSR  R5,FUNCT
1721 006024 000005  .WORD 5
1722 006026 105777 017554          TSTB 2RPCS          ;WAIT FOR DONE AFTER READ
1723 006032 100375          BPL  -4
1724 006034 005777 017546          TST  2RPCS          ;ANY ERRORS?
1725 006040 100002          BPL  3$             ;BRANCH IF NO
1726 006042 104400          HLT
1727 006044 000413          BR   5$
1728 006046 027767 017616 017570  3$:  CMP  2INBUF,CYLINDER ;COMPARE DATA READ AGAINST CYLINDER

```

1729	006054	001407				BEQ	5\$		; BRANCH IF EQUAL
1730	006056	016767	017562	011672		MOV	CYLINDER,EXPS		; CORRECT DATA
1731	006054	017767	017600	011666		MOV	INBUF,RECS		; INCORRECT DATA
1732	006072	104401				HLT	+1		; DATA COMPARE ERROR-PROBABLY WENT
1733									; TO THE WRONG CYLINDER
1734	006074	032777	004000	017524	5\$:	BIT	#B11,DRPDS		; SEEK INCOMPLETE ERROR?
1735	006102	001412				BEQ	2\$		; BRANCH IF NO
1736	006104	112777	000015	017474		MOVB	#15,DRPCS		; ISSUE A HOME COMMAND
1737	006112	105777	017470			TSTB	DRPCS		; WAIT FOR DONE
1738	006116	100375				BPL	-4		
1739	006120	032777	100000	017500	4\$:	BIT	#B15,DRPDS		; WAIT FOR UNIT READY
1740	006126	001774				BEQ	4\$		
1741	006130	012767	006012	172644	2\$:	MOV	#ADT32,LAD		; SETUP LOOP
1742	006136	104000				SCOPE			
1743	006140	005767	017546			TST	WORK4		; INC - DEC FLAG
1744	006144	100411				BMI	1\$		
1745	006146	005267	017536			INC	WORK3		; UPDATE LOW COUNT
1746	006152	016767	017530	017454		MOV	WORK2,CYLINDER		
1747	006160	052767	100000	017524		BIS	#B15,WORK4		; SET DECREMENT FLAG
1748	006166	000711				BR	ADT32		
1749	006170	005367	017512		1\$:	DEC	WORK2		; DECREMENT HIGH COUNT
1750	006174	005067	017512			CLR	WORK4		; CLEAR FLAG
1751	006200	016767	017504	017436		MOV	WORK3,CYLINDER		
1752	006206	005767	017474			TST	WORK2		; DONE YET
1753	006212	001277				BNE	ADT32		; BRANCH-NO
1754	006214	032737	002000	177570		BIT	#B10,DRSWR		; LOOP ON TEST?
1755	006222	001402				BEQ	WRCK		; NO
1756	006224	000167	177366			JMP	RAOT3		; YES

1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812

.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.

```
WRCK:  MOV    #3,TESTNO
      JSR    RS,PRINT$      ;PRINT MESSAGE
      MES6
1771:  MOV    TESTNO,TTY
      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
      CLR    MEX            ;CLEAR EXTENDED MEMORY BITS IN CONTROLER
RWRCK: JSR    PC,INIT       ;INITIALIZE
      JSR    RS,DSKNOS      ;SELECT UNIT
      CLR    CYLINDER
      CLR    DMA
      CLR    RO            ;PATTERN FLAG
      MOV    #1,R1         ;STARTING PATTERN
      MOV    #OUTBUF,BUF   ;SETUP OUTPUT BUFFER
      MOV    #400,WRDCT    ;SETUP WORDCOUNT
23$:  CLR    R2
1$:   MOV    R1,OUTBUF(R2) ;GENERATE TEST PATTERN
      TST   (R2)+          ;UPDATE MODIFIER
      CMP   #400,R2       ;HAS BUFFER BEEN FILLED?
      BNE  1$            ;BRANCH IF NO
      JSR  RS,FUNCT      ;WRITE PATTERN
.WORD 3
1789:  TSTB   @RPCS        ;WAIT FOR DONE
      BPL   -4
1791:  TST   @RPCS        ;ANY DEVICE ERRORS?
      BPL   2$          ;BRANCH IF NO
      HLT   ;ERROR AFTER WRITING ONE SECTOR
1795:  BR    3$
2$:   JSR    RS,FUNCT    ;WRITE CHECK THE DATA
.WORD 7
1797:  TSTB   @RPCS        ;WAIT FOR DONE
      BPL   -4
1799:  TST   @RPCS        ;ANY DEVICE ERRORS?
      BPL   4$          ;BRANCH IF NO
      HLT   ;ERROR AFTER WRITE CHECK OPERATION
1802:  JSR    RS,PRINT$    ;PRINT MESSAGE
      MES10
1804:  MOV    OUTBUF,TTY
      JSR    PC,PRINT$    ;TYPE LOCATION WITH LEADING ZEROS
      BR    3$
4$:   TST   RO            ;ARE WE FLOATING A ONE?
      BEQ  20$          ;BRANCH IF YES
      CLR  R2            ;FILL BUFFER WITH ONES
21$:  MOV    #177777,OUTBUF(R2)
      TST  (R2)+
      CMP  #400,R2
```

1813	006460	001371			BNE	21\$	
1814	006462	000407			BR	22\$	
1815	006464	005002			20\$: CLR	R2	
1816	006466	005062	025730		5\$: CLR	OUTBUF(R2)	;CLEAR OUTPUT BUFFER
1817	006472	005722			TST	(R2)+	
1819	006474	022702	000400		CMP	#400,R2	;ENTIRE BUFFER CLEAR?
1919	006500	001372			BNE	5\$	;BRANCH IF NO
1820	006502	004567	006426		22\$: JSR	R5,FUNCT	;WRITE CHECK THE DATA AND
1821	006506	000007			.WORD	7	;EXPECT AN ERROR
1822	006510	105777	017072		TSTB	DRPCS	;WAIT FOR DONE
1823	006514	100375			BPL	.-4	
1824	006516	032777	000010	017100	BIT	#B3,DRPER	;IS WRITE CHECK ERROR SET?
1825	006524	001011			BNE	6\$	;BRANCH IF YES
1826	006526	104400			HLT		;WRITE CHECK ERROR DID NOT SET
1827	006530	004567	172530		JSR	R5,PRINTS	;PRINT MESSAGE
1828	006534	024042			MES10		
1829	006536	010167	173020		MOV	R1,TTY	
1830	006542	004767	172576		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS
1831	006546	000405			BR	3\$	
1832	006550	005777	017032		6\$: TST	DRPCS	;DID ERROR FLAG SET?
1833	006554	100402			BMI	3\$	;BRANCH IF YES
1834	006556	104400			HLT		;ERROR FLAG DID NOT SET AFTER WRITE CHECK ERROR
1835	006560	000400			BR	3\$	
1836	006562	0127E7	006324	172212	3\$: MOV	#23\$,LAD	;SETUP LOOP ADDR
1837	006570	104000			SCOPE		
1838	006572	005700			TST	R0	;ARE WE FLOATING A ONE?
1839	006574	001013			BNE	9\$	;BRANCH IF NO
1840	006576	000241			CLC		
1841	006600	006101			ROL	R1	;ROTATE PATTERN
1842	006602	103402			BCS	10\$	;BRANCH IF COMPLETE
1843	006604	000167	177514		JMP	23\$	
1844	006610	012700	000001		10\$: MOV	#1,R0	;SET PATTERN FLAG
1845	006614	012701	077777		MOV	#077777,R1	;SET NEW PATTERN IN R1
1846	006620	000167	177500		JMP	23\$	
1847	006624	000241			9\$: CLC		
1848	006626	006201			ASR	R1	;ROTATE FLOATING ZERO PATTERN
1849	006630	052701	100000		BIS	#B15,R1	
1850	006634	103002			BCC	PATFIL	;HAS ZERO BEEN FLOATED
1851	006636	000167	177462		JMP	23\$	;JUMP IF NO
1852							
1853							
1854							
1855							
1856							
1857							
1858							
1859							
1860							
1861	006642	012701	025730		PATFIL: MOV	#OUTBUF,R1	
1862	006646	012700	177777		MOV	#177777,R0	
1863	006652	012702	000400		MOV	#400,R2	
1864	006656	010021			1\$: MOV	R0,(R1)+	;GENERATE ALL ONES PATTERN
1865	006660	005302			DEC	R2	
1866	006662	001375			BNE	1\$	
1867	006664	004567	006244		JSR	R5,FUNCT	;WRITE SECTOR WITH ONES
1868	006670	000003			.WORD	3	

;CHECK THE ABILITY OF THE RP11C TO CLEAR THE REMAINDER OF A SECTOR  
 ;ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND  
 ;THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN  
 ;READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND  
 ;THE REST SHOULD BE ZEROS.



1869	006672	105777	016710		TSTB	DRPCS			;WAIT FOR DONE
1870	006676	100375			BPL	-4			
1871	006700	005777	016702		TST	DRPCS			;ANY DEVICE ERRORS
1872	006704	100002			BPL	25			;BRANCH IF NO
1873	006706	104400			HLT				;ERROR AFTER WRITING ONE SECTOR ALL 1'S
1874	006710	000473			BR	35			
1875	006712	112767	000002	016722	25:	MOV	#2,WRDCT		;SETUP FOR TWO WORD WRITE
1876	006727	004567	006210		JSR	RS,FUNCT			;WRITE TWO WORD
1877	006724	000003				3		.WORD	
1878	006726	105777	016654		TSTB	DRPCS			;WAIT FOR DONE
1879	006732	100375			BPL	-4			
1880	006734	005777	016646		TST	DRPCS			;ANY ERRORS?
1881	006740	100002			BPL	45			;BRANCH IF NO
1882	006742	104400			HLT				;ERROR ON ONE WORD WRITE
1883	006744	000455			BR	35			
1884	006746	012767	000400	016656	45:	MOV	#400,WRDCT		;SETUP WORD COUNT
1885	006754	004567	006154		JSR	RS,FUNCT			;READ SECTOR
1886	006760	000005				5		.WORD	
1887	006762	105777	016620		TSTB	DRPCS			;WAIT FOR DONE
1888	006766	100375			BPL	-4			
1889	006770	005777	016612		TST	DRPCS			;ANY ERRORS
1890	006774	100006			BPL	55			;BRANCH IF NO
1891	006776	104400			HLT				;ERROR AFTER READING ONE SECTOR
1892	007000	032777	040000	016600		#814,DRPCS			;WAS IT A DATA ERROR?
1893	007006	001401			BEQ	55			;BRANCH IF YES
1894	007010	000433			BR	35			
1895	007012	022767	177777	016710	55:	CMP	#177777,OUTBUF		;COMPARE FIRST WORD SHOULD BE ONES
1896	007020	001410			BEQ	65			;BRANCH IF OK
1897	007022	012767	177777	010726		MOV	#177777,EXPS		
1898	007030	016767	016674	010722		MOV	OUTBUF,RECS		
1899	007036	104401			HLT	+1			;DATA COMPARE ERROR ON FIRST
1900	007040	000417			BR	35			;WORD READ
1901	007042	012700	025734		65:	MOV	#OUTBUF+4,R0		
1902	007046	012701	000374			MOV	#374,R1		
1903	007052	005720			95:	TST	(R0)+		;REMAINDER OF SECTOR SHOULD BE CLEAR
1904	007054	001003				BNE	75		;BRANCH IF NO
1905	007056	005301				DEC	R1		
1906	007060	001374				BNE	85		
1907	007062	000406				BR	35		
1908	007064	016067	177776	010666	75:	MOV	-2(R0),RECS		
1909	007072	005067	010660			CLR	EXPS		
1910	007076	104401				HLT	+1		;DATA FOUND IN AREA OF SECTOR
1911									;WHICH SHOULD HAVE BEEN CLEARED
1912									;BY A ONE WORD WRITE
1913	007100	012767	006642	171674	35:	MOV	#PATFIL,LAD		;SET UP LOOP ADDR
1914	007106	104000				SCOPE			
1915									
1916									
1917									
1918									
1919									
1920									
1921									
1922									
1923									
1924	007110	005067	016532		EOPTST:	CLR	DMA		;CLEAR DISK ADDRESS

;CHECK THE SETTING OF EOP WHEN TRYING TO WRITE BEYOND  
 ;THE LIMITS OF THE PACK. THE FIRST SECTOR OF THE PACK IS  
 ;WRITTEN WITH ZEROS. THEN A TWO SECTOR WRITE OF ALL  
 ;ONE'S IS ISSUED FOR CYLINDER 625, HEAD 23, AND SECTOR 1!.  
 ;EOP AND ERROR BITS SHOULD SET. THE FIRST SECTOR OF THE  
 ;PACK IS CHECKED TO MAKE SURE IT IS STILL ZERO.

1925	007114	005067	016524			CLR	CYLINDER	
1926	007120	012767	000400	016514		MOV	#400,WRDCT	;SET WORDCOUNT TO ONE SECTOR
1927	007126	012767	025730	016520		MOV	#OUTBUF,BUF	;SETUP OUTPUT BUFFER
1928	007134	005001				CLR	R1	
1929	007136	005061	025730		15:	CLR	OUTBUF(R1)	;CLEAR THE OUTPUT BUFFER
1930	007142	005721				TST	(R1)+	
1931	007144	022701	000400			CMP	#400,R1	
1932	007150	001372				BNE	15	
1933	007152	004567	005756			JSR	R5,FUNCT	;WRITE SECTOR ZERO WITH ZEROS
1934	007156	000003			.WORD	3		
1935	007160	105777	016422			TSTB	2RPCS	;WAIT FOR DONE
1936	007164	100375				BPL	-4	
1937	007166	005777	016414			TST	2RPCS	;ANY DEVICE ERRORS
1938	007172	100002				BPL	25	;BRANCH IF NO
1939	007174	104400				HLT		;ERROR AFTER WRITING SECTOR ZERO WITH ZEROS
1940	007176	000502				BR	35	
1941	007200	012767	001000	016434	25:	MOV	#1000,WRDCT	;SET WORDCOUNT EQUAL TO TWO SECTORS
1942	007206	012767	000625	016430		MOV	#625,CYLINDER	;SELECT CYLINDER 625
1943	007214	012767	000011	016424		MOV	#11,DMA	;SELECT RECTOR 11
1944	007222	112767	000023	016417		MOVB	#23,DMA+1	;SELECT HEAD 23
1945	007230	012702	177777			MOV	#177777,R2	
1946	007234	005001				CLR	R1	
1947	007236	010261	025730		45:	MOV	R2,OUTBUF(R1)	;SET OUTPUT BUFFER TO ONES
1948	007242	005721				TST	(R1)+	
1949	007244	022701	001000			CMP	#1000,R1	
1950	007250	001372				BNE	45	
1951	007252	004567	005656			JSR	R5,FUNCT	;ISSUE TWO SECTOR WRITE TO
1952	007256	000003			.WORD	3		;CYLINDER 625, HEAD 23, AND SECTOR 11
1953	007260	105777	016322			TSTB	2RPCS	;WAIT FOR DONE
1954	007264	100375				BPL	-4	
1955	007266	032777	000002	016330		BIT	#81,2RPER	;DID EOP ERROR FLAG SET?
1956	007274	001002				BNE	55	;BRANCH IF SET
1957	007276	104400				HLT		;EOP ERROR FLAG DID NOT SET WHEN
1958	007300	000441				BR	35	;WRITE OPERATOR EXCEEDS THE PACK
1959	007302	032777	100000	016276	55:	BIT	#815,2RPCS	;DID THE ERROR FLAG SET?
1960	007310	001002				BNE	65	;BRANCH IF SET
1961	007312	104400				HLT		;ERROR DID NOT SET AFTER GENERATING
1962	007314	000433				BR	35	;EOP
1963	007316	012767	000002	016316	65:	MOV	#2,WRDCT	
1964	007324	005067	016316			CLR	DMA	;CLEAR THE DISK ADDRESS
1965	007330	005067	016310			CLR	CYLINDER	
1966	007334	004567	005574			JSR	R5,FUNCT	;READ THE FIRST SECTOR OF THE PACK
1967	007340	000005			.WORD	5		;AND EXPECT TO FIND ZEROS
1968	007342	105777	016240			TSTB	2RPCS	;WAIT FOR READY
1969	007346	100375				BPL	-4	
1970	007350	005777	016232			TST	2RPCS	;WERE THERE ANY ERRORS?
1971	007354	100002				BPL	75	;BRANCH IF NO
1972	007356	104400				HLT		;ERROR ENCOUNTERED ON 2 WORD READ
1973	007360	000411				BR	35	;OF FIRST SECTOR ON THE PACK
1974	007362	016767	016342	010370	75:	MOV	OUTBUF,RECS	;GET FIRST WORD OF BUFFER
1975	007370	005767	010364			TST	RECS	;DOES 1ST SECTOR STILL CONTAIN ZEROS?
1976	007374	001403				BEQ	35	;BRANCH IF YES
1977	007376	005067	010354			CLR	EXPS	
1978	007402	104401				HLT	+1	;CONTENTS OF THE FIRST SECTOR OF THE
1979								;PACK CHANGED AFTER FORCING EOP
1980								;ERROR, OPERATION PROBABLY

```

1981                                     ;WRAPPED AROUND.
1982 007404 012767 007110 171370 3$:  MOV  #EOPTST,LAD
1983 007412 104000                      SCOPE
1984
1985
1986                                     ;CHECK THE ABILITY OF THE RP11C TO GENERATE A
1987                                     ;PROGRAM ERROR IF A COMMAND IS ISSUED WHILE THE
1988                                     ;CONTROLLER IS BUSY.
1989
1990 007414 012777 000001 016164 LOKOUT: MOV  #1,ARPC$      ;CLEAR THE CONTROLLER
1991 007422 005067 016216          CLR  CYLINDER      ;CLEAR THE DISK ADDRESS
1992 007426 005067 016214          CLR  DMA
1993 007432 012767 002000 016202  MOV  #2000,WRDCT    ;SETUP WORDCOUNT
1994 007440 004567 005470          JSR  RS,FUNCT     ;ISSUE A READ COMMAND TO GET THE
1995 007444 000005          .WORD  S                    ;CONTROLLER BUSY
1996 007446 012700 000100          MOV  #100,RO
1997 007452 005300          1$:  DEC  RO              ;ALLOW THE OPERATION TO PROCEED AWHILE
1998 007454 001376          BNE  1$
1999 007456 112777 000005 016122  MOVB #5,ARPC$      ;ISSUE READ COMMAND WHILE BUSY
2000 007464 105777 016116          TSTB ARPC$        ;WAIT FOR DONE
2001 007470 100375          BPL  -4
2002 007472 032777 002000 016124  BIT  #B10,ARPER   ;DID PROGRAM ERROR SET?
2003 007500 001002          BNE  2$          ;BRANCH IF SET
2004 007502 104400          HLT                    ;PROGRAM ERROR DID NOT SET WHEN A
2005                                     ;READ COMMAND WAS ISSUED WHILE
2006                                     ;THE DEVICE WAS BUSY
2007 007504 000405          BR   3$
2008 007506 032777 100000 016072 2$:  BIT  #B15,ARPC$   ;DID THE ERROR FLAG SET?
2009 007514 001001          BNE  3$          ;BRANCH IF SET
2010 007516 104400          HLT                    ;ERROR FLAG DID NOT SET AFTER
2011                                     ;PROGRAM ERROR
2012 007520 012767 007414 171254 3$:  MOV  #LOKOUT,LAD
2013 007526 104000                      SCOPE
2014
2015
2016                                     ;UNFORMAT THE FIRST SECTOR ON THE PACK. THEN READ IT BACK
2017                                     ;AND VERIFY THAT READ AND WRITE HEADER OPERATIONS WILL
2018                                     ;TRANSFER DATA CORRECTLY. NOW THAT THE HEADER IS MIS FORMATTED,
2019                                     ;ISSUE A WRITE COMMAND TO SECTOR ZERO. THIS SHOULD RESULT
2020                                     ;IN SETTING HEADER NOT FOUND. THEN REFORMAT SECTOR ZERO.
2021
2022 007530 012777 000001 016050 HNFCK: MOV  #1,ARPC$      ;CLEAR THE CONTROLLER
2023 007536 005067 016102          CLR  CYLINDER      ;CLEAR DISK ADDR
2024 007542 005067 016100          CLR  DMA
2025 007546 012767 025730 016100  MOV  #OUTBUF,BUF   ;SETUP BUFFER ADDR
2026 007554 012700 025730          MOV  #OUTBUF,RO
2027 007560 012720 000001          MOV  #1,(RO)+
2028 007564 012720 000001          MOV  #1,(RO)+
2029 007570 012720 000001          MOV  #1,(RO)+
2030 007574 012767 000003 016040  MOV  #3,WRDCT     ;LOAD WORDCOUNT
2031 007602 004567 005326          JSR  RS,FUNCT     ;ISSUE WRITE HEADER COMMAND
2032 007606 014003          .WORD 14003      ;TO MISFORMAT SECTOR ZERO
2033 007610 105777 015772          TSTB ARPC$        ;WAIT FOR READY
2034 007614 100375          BPL  -4
2035 007616 005777 015764          TST  ARPC$        ;ANY ERRORS?
2036 007622 100002          BPL  1$          ;BRANCH IF NO
  
```

2037	007624	104400				HLT			; ERROR AFTER ISSUING A WRITE
2038									; FORMAT COMMAND TO SECTOR ZERO
2039	007626	000440				BR	2\$		
2040	007630	004567	005300		1\$:	JSR	R5, FUNCT		; ISSUE READ HEADER COMMAND
2041	007634	014005			.WORD	14005			; TO SECTOR ZERO
2042	007636	105777	015744			TSTB	@RPCS		; WAIT FOR READY
2043	007642	100375				BPL	-4		
2044	007644	005777	015736			TST	@RPCS		; ANY ERRORS?
2045	007650	100005				BPL	3\$		; BRANCH IF NO
2046	007652	104400				HLT			; ERROR WHILE READING THE HEADER
2047									; ON SECTOR ZERO
2048	007654	032777	040000	015724		BIT	#B14, @RPCS		; HARD ERROR?
2049	007662	001022				BNE	2\$		; BRANCH IF YES
2050	007664	012767	000001	010064	3\$:	MOV	#1, EXP\$		; EXPECTED RESULT
2051	007672	005000				CLR	RO		
2052	007674	026760	010056	025730	5\$:	CMP	EXP\$, OUTBUF(RO)		; CHECK DATA READ BACK
2053	007702	001006				BNE	4\$		; BRANCH ON NON COMPARE
2054	007704	062700	000002			ADD	#2, RO		; UPDATE MODIFIER
2055	007710	022700	000006			CMP	#6, RO		; END OF BUFFER?
2056	007714	001367				BNE	5\$		
2057	007716	000404				BR	2\$		
2058	007720	016067	025730	010032	4\$:	MOV	OUTBUF(RO), RECS		; GET BAD DATA
2059	007726	104403				HLT	+3		; DATA DID NOT VERIFY AFTER READING
2060									; THE HEADER OF SECTOR ZERO
2061	007730	012767	007530	171044	2\$:	MOV	#HNFCK, LAD		
2062	007736	104000				SCOPE			
2063	007740	004567	005170		HNF1:	JSR	R5, FUNCT		; ISSUE WRITE TO SACTOR ZERO
2064	007744	000003			.WORD	3			
2065	007746	005000				CLR	RO		
2066	007750	005200			2\$:	INC	RO		; TIMEOUT SETTING OF DONE
2067	007752	105777	015630			TSTB	@RPCS		; DID DONE SET?
2068	007756	100414				BMI	1\$		
2069	007760	005237	025714			INC	@#CYLA		
2070	007764	005237	025714			INC	@#CYLA		
2071	007770	005337	025714			DEC	@#CYLA		
2072	007774	005337	025714			DEC	@#CYLA		
2073	010000	005700				TST	RO		; TIMEOUT?
2074	010002	001362				BNE	2\$		; BRANCH IF NO
2075	010004	104400				HLT			; TIMED OUT WHILE WAITING FOR DONE
2076									; AFTER TRYING TO FORCE HEADER NOT FOUND
2077	010006	000427				BR	3\$		
2078	010010	032777	010000	015610	1\$:	BIT	#B12, @RPDS		; DID HEADER NOT FOUND SET?
2079	010016	001002				BNE	6\$		; BRANCH IF YES
2080	010020	104400				HLT			; HEADER NOT FOUND DID NOT SET
2081	010022	000421				BR	3\$		
2082	010024	032777	040000	015554	6\$:	BIT	#B14, @RPCS		; DID HARD ERROR SET?
2083	010032	001002				BNE	4\$		; BRANCH IF YES
2084	010034	104400				HLT			; HARD ERROR NOT SET AFTER HNF
2085	010036	000413				BR	3\$		
2086	010040	032777	100000	015540	4\$:	BIT	#B15, @RPCS		; DID THE ERROR BIT SET?
2087	010046	001002				BNE	5\$		; BRANCH IF YES
2088	010050	104400				HLT			; ERROR DID NOT SET AFTER HNF
2089	010052	000405				BR	3\$		
2090	010054	032777	000001	015542	5\$:	BIT	#B0, @RPER		; DID DISK ERROR SET?
2091	010062	001001				BNE	3\$		; BRANCH IF YES
2092	010064	104400				HLT			; DISK ERROR DID NOT SET AFTER HNF

```

2093 010066 012767 007740 170706 3$: MOV #HNF1,LAD
2094 010074 104000 SCOPE
2095 010076 005000 HNF2: CLR RO
2096 010100 005060 025730 1$: CLR OUTBUF(RO) ;CLEAR BUFFER
2097 010104 062700 000002 ADD #2,RO ;UPDATE MODIFIER
2098 010110 022700 000006 CMP #6,RO ;ENTIRE BUFFER CLEARED?
2099 010114 001371 BNE 1$ ;BRANCH IF NO
2100 010116 012767 000003 015516 MOV #3,WRDCT ;LOAD WORD COUNT
2101 010124 004567 705004 JSR R5,FUNCT ;ISSUE WRITE HEADER COMMAND
2102 010130 014003 .WORD 14003
2103 010132 105777 015450 TSTB @RPCS ;WAIT FOR READY
2104 010136 100375 BPL -4
2105 010140 005777 015442 TST @RPCS ;ANY ERRORS?
2106 010144 100001 BPL 2$
2107 010146 104400 HLT ;ERROR WHILE TRYING TO FORMAT SECTOR ZERO
2108 010150 012767 010076 170624 2$: MOV #HNF2,LAD
2109 010156 104000 SCOPE
2110
2111 ;ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A WRITE
2112 ;COMMAND WHILE THE HEADS ARE STILL MOVING. THE RP11C SHOULD
2113 ;HOLD THE WRITE COMMAND TILL THE SEEK IS COMPLETE.
2114
2115 010160 012777 000001 015420 SKTST: MOV #1,@RPCS ;CLEAR THE CONTROLLER
2116 010166 004567 006376 JSR R5,DSKNOS ;SELECT THE UNIT
2117 010172 005077 015420 CLR @RPCA ;CLEAR THE CYLINDER ADDR REGISTER
2118 010176 005077 015416 CLR @RPDA ;CLEAR THE DISK ADDR REGISTER
2119 010202 112777 000011 015376 MOVB #11,@RPCS ;ISSUE A SEEK TO ZERO
2120 010210 105777 015372 TSTB @RPCS ;WAIT FOR DONE
2121 010214 100375 BPL -4
2122 010216 032777 100000 015402 1$: BIT #B15,@RPDS ;WAIT FOR UNIT READY
2123 010224 001774 BEQ 1$
2124 010226 012777 000300 015362 MOV #300,@RPCA ;SELECT CYLINDER 300
2125 010234 112777 000011 015344 MOVB #11,@RPCS ;ISSUE SEEK TO CYLINDER 300
2126 010242 105777 015340 TSTB @RPCS ;WAIT FOR DONE
2127 010246 100375 BPL -4
2128 010250 012777 177777 015334 MOV #-1,@RPWC ;SETUP WORD COUNT
2129 010256 112777 000003 015322 MOVB #3,@RPCS ;ISSUE A WRITE WHILE HEADS ARE MOVUNG
2130 010264 105777 015316 TSTB @RPCS ;WAIT FOR DONE
2131 010270 100375 BPL -4
2132 010272 005777 015310 TST @RPCS ;ANY ERRORS?
2133 010276 100001 BPL 2$ ;BRANCH IF NO
2134 010300 104400 HLT ;ERROR FOUND AFTER ISSUING A WRITE
2135 ;COMMAND WHILE THE HEADS ARE STILL
2136 ;MOVING AFTER A SEEK.
2137 010302 032777 004000 015316 2$: BIT #B11,@RPDS ;SEEK INCOMPLETE ERROR?
2138 010310 001412 BEQ 3$ ;BRANCH IF NO
2139 010312 112777 000015 015266 MOVB #15,@RPCS ;ISSUE A HOME COMMAND
2140 010320 105777 015262 TSTB @RPCS ;WAIT FOR DONE
2141 010324 100375 BPL -4
2142 010326 032777 100000 015272 4$: BIT #B15,@RPDS ;WAIT FOR UNIT READY
2143 010334 001774 BEQ 4$
2144 010336 012767 010160 170436 3$: MOV #SKTST,LAD
2145 010344 104000 SCOPE
2146
2147 ;CHECK THE ABILITY OF CLEAR TO TERMINATE AN OPERATION
2148 ;AND SET READY.

```

2149											
2150	010346	004567	006216			CLRTST:	JSR	R5,DSKNOS		;SELECT THE UNIT	
2151	010352	005067	015270				CLR	DMA		;CLEAR DISK ADDR	
2152	010356	005067	015262				CLR	CYLINDER			
2153	010362	012767	000001	015252			MOV	#1,WRDCT		;SETUP WORD COUNT	
2154	010370	012767	025730	015256			MOV	#OUTBUF, BUF		;SETUP BUFFER ADDR	
2155	010376	004567	004532				JSR	R5,FUNCT		;ISSUE A WRITE COMMAND	
2156	010402	000003				.WORD	3				
2157	010404	012700	000100				MOV	#100,RO			
2158	010410	005300				1\$:	DEC	RO		;WAIT A WHILE THEN ISSUE A	
2159	010412	001376					BNE	1\$		;CLEAR COMMAND	
2160	010414	112777	000001	015164			MOVB	#1,DRPCS		;CLEAR THE CONTROLLER	
2161	010422	105777	015160				TSTB	DRPCS		;IS READY SET?	
2162	010426	100401					BMI	2\$		;BRANCH IF YES	
2163	010430	104400					HLT			;READY DID NOT SET AFTER ISSUING	
2164										;A CLEAR COMMAND DURING A WRITE.	
2165	010432	000005				2\$:	RESET				
2166	010434	012767	010346	170340			MOV	#CLRTST,LAD			
2167	010442	104000					SCOPE				
2168	010444	016767	015162	015230			MOV	FLAG,WORK		;GET UNIT NUMBER	
2169	010452	000241					CLC				
2170	010454	006067	015222				ROR	WORK			
2171	010460	006067	015216				ROR	WORK			
2172	010464	000367	015212				SWAB	WORK		;JUSTIFY UNIT NUMBER	
2173	010470	042767	174377	015204			BIC	#174377,WORK		;CLEAR UNWANTED BITS	
2174	010476	016777	015200	015102			MOV	WORK,DRPCS		;LOAD UNIT NUMBER	
2175	010504	005777	015116				TST	DRPDS		;WAIT FOR UNIT READY	
2176	010510	100375					BPL	.-4			
2177	010512	032737	002000	177570			BIT	#B10,DRSWR		;LOOP ON TEST?	
2178	010520	001402					BEQ	MEMTST		;BRANCH IF NO	
2179	010522	000167	175534				JMP	RWRCK			

.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.

2180  
2181  
2182  
2183  
2184  
2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235

010526 012767 000004 015142  
010534 004567 170524  
010540 023740  
010542 016767 015130 171012  
010550 004767 170602  
010554 012767 177700 015104  
010562 032767 000500 015042  
010570 001003  
010572 012767 177750 015066  
010600 005067 011146  
010604 004767 006032  
010610 032767 000100 015014  
010616 001520  
010620 016700 011124  
010624 162700 025730  
010630 000241  
010632 006000  
010634 042700 000001  
010640 010067 014776  
010644 012702 025730  
010650 012703 025730  
010654 010322  
010656 005723  
010660 020267 011064  
010664 101773  
010666 012767 025730 014760  
010674 005067 014746  
010700 005067 014740  
010704 004567 004224  
010710 000003  
010712 105777 014670  
010716 100375  
010720 005777 014662  
010724 100002  
010726 104400  
010730 000446  
010732 016700 014704  
010736 012701 025730  
010742 005021  
010744 005300  
010746 001375  
010750 004567 004160  
010754 000005  
010756 105777 014624  
010762 100375

MEMTST: MOV #4,TESTNO  
JSR R5,PRINT\$ ;PRINT MESSAGE  
MES6  
MOV TESTNO,TTY  
JSR PC,PRINT\$ ;TYPE LOCATION-SUPRESS ZEROS  
MOV #-100,PASSC ;SETUP ITERATION COUNT NO MEM. MANAG.  
BIT #500,FLAG ;8K WITH MEM. MANAG. OR NO MEM. MANAG.?  
BNE 1\$ ;BR IF TRUE  
MOV #-30,PASSC ;IF MEMORY MANAGEMENT DO ONLY 24 PASSES  
1\$: CLR MEX ;CLEAR DRIVE EXTENDED MEMORY BITS  
RMEMT: JSR PC,INIT ;INITIALIZE  
BIT #86,FLAG ;MEMORY MANAGEMENT?  
BEQ RMMEMT ;YES GO DO MEMORY MANAGEMENT TEST  
MEMBK: MOV MEMSIZ,R0 ;GET TOP OF CORE  
SUB #OUTBUF,R0 ;DETERMINE SIZE OF BUFFER IN BYTES  
CLC  
ROR R0 ;CONVERT TO WORDS  
BIC #1,R0 ;KEEP NUMBER EVEN  
MOV R0,WRDCT ;SAVE WORD COUNT OF TRANSFER  
7\$: MOV #OUTBUF,R2  
MOV #OUTBUF,R3 ;GENERATE A PATTERN SO THAT EACH  
1\$: MOV R3,(R2)+ ;LOCATION CONTAINS ITS ADDRESS  
TST (R3)+  
CMP R2,MEMSIZ ;HAS ENTIRE PATTERN BEEN GENERATED?  
BLOS 1\$ ;BRANCH IF NO  
MOV #OUTBUF,BUF ;SET UP BUFFER ADDR  
CLR DMA  
CLR CYLINDER  
JSR R5,FUNCT ;WRITE ADDRESS PATTERN  
.WORD 3  
TSTB @RPCS ;WAIT FOR DONE  
BPL -4  
TST @RPCS ;ANY ERRORS?  
BPL 2\$ ;BRANCH IF NO  
HLT ;ERROR AFTER WRITING ADDR PATTERN  
BR 3\$  
2\$: MOV WRDCT,R0  
MOV #OUTBUF,R1  
10\$: CLR (R1)+ ;CLEAR THE BUFFER  
DEC R0  
BNE 10\$  
JSR R5,FUNCT ;READ ADDRESS PATTERN  
.WORD 5  
TSTB @RPCS ;WAIT FOR DONE  
BPL -4

```

2236 010764 005777 014616          TST      @RPCS          ;ANY ERRORS?
2237 010770 100006          BPL      4$            ;BRANCH IF NO
2238 010772 104400          HLT                                ;ERROR AFTER READING ADDR PATTERN
2239 010774 032777 040000 014604  BIT      #B14,@RPCS    ;IS THIS A DATA ERROR?
2240 011002 001401          BEQ      4$            ;BRANCH IF YES
2241 011004 000420          BR       3$
2242 011006 012702 025730          4$:     MOV      #OUTBUF,R2
2243 011012 010203          MOV      R2,R3
2244 011014 020322          6$:     CMP      R3,(R2)+  ;COMPARE THE ADDR PATTERN
2245 011016 001005          BNE      5$            ;IS DATA CORRECT?
2246 011020 005723          TST      (R3)+        ;BRANCH IF NO
2247 011022 020267 010722          CMP      R2,MEMSIZ    ;IS ENTIRE BUFFER VERIFIED?
2248 011026 101772          BLOS    6$            ;BRANCH IF NO
2249 011030 000406          BR       3$
2250 011032 010367 006720          5$:     MOV      R3,EXP$
2251 011036 016267 177776 006714  MOV      -2(R2),REC$
2252 011044 104401          HLT      +1
2253 011046 012767 010644 167726  3$:     MOV      #7$,LAD   ;COMPARE ERROR UTILIZING ADDR PATTERN
2254 011054 104000          SCOPE
2255 011056 000527          BR       NONEX        ;SETUP LOOP ADD
2256                                     ;GO DO REST OF TEST
2257                                     ;THIS PORTION IS FOR MEMORY MANAGEMENT
2258
2259 011060 032767 000400 014544  RMMEMT: BIT      #B8,FLAG  ;BK MACHINE?
2260 011066 001254          BNE      MEMBK        ;BR IF YES
2261 011070 004767 010670          JSR      PC,PARINT    ;INITIALIZE KIPARS
2262 011074 004767 010754          1$:     JSR      PC,PARINC ;INCREMENT KIPARS
2263 011100 004767 011000          JSR      PC,PARREG    ;GENERATE "MEX", "BUF", "WRDCT"
2264 011104 012700 040000          5$:     MOV      #40000,R0 ;STARTING ADDRESS OF BUFFER
2265 011110 016701 014526          MOV      WRDCT,R1
2266 011114 010010          2$:     MOV      R0,(R0)  ;GET BUFFER SIZE
2267 011116 005720          TST      (R0)+        ;MOVE ADDRESS TO LOCATION
2268 011120 005301          DEC      R1           ;BUMP LOCATION
2269 011122 001374          BNE      2$          ;DONE?
2270 011124 005067 014516          CLR      DMA          ;BR IF NO
2271 011130 005067 014510          CLR      CYLINDER
2272 011134 004567 003774          JSR      R5,FUNCT    ;WRITER ADDRESS PATTERN
2273 011140 000003          .WORD   3
2274 011142 105777 014440          TSTB    @RPCS        ;WAIT FOR DONE
2275 011146 100375          BPL      -4
2276 011150 005777 014432          TST      @RPCS        ;ANY ERRORS?
2277 011154 100005          BPL      3$          ;BR IF NO
2278 011156 104400          HLT                                ;ERROR WRITTING ADDRESS PATTERN
2279 011160 012767 011104 167614  MOV      #5$,LAD     ;SETUP LOOP ADDRESS
2280 011166 104000          SCOPE
2281 011170 012700 040000          3$:     MOV      #40000,R0 ;START OF BUFFER
2282 011174 016701 014442          MOV      WRDCT,R1    ;BUFFER SIZE
2283 011200 005020          4$:     CLR      (R0)+   ;CLEAR BUFFER
2284 011202 005301          DEC      R1           ;DONE?
2285 011204 001375          BNE      4$          ;BR IF NO
2286 011206 004567 003722          6$:     JSR      R5,FUNCT ;READ THE ADDRESS PATTERN
2287 011212 000005          .WORD   5
2288 011214 105777 014366          TSTB    @RPCS        ;WAIT FOR DONE
2289 011220 100375          BPL      -4
2290 011222 005777 014360          TST      @RPCS        ;ANY ERRORS
2291 011226 100011          BPL      7$          ;BR IF NO

```



2292	011230	104400				HLT			;DISK ERROR
2293	011232	032777	040000	014346		BIT	#B14,DRPCS		;DATA ERROR?
2294	011240	001404				BEQ	7\$		;BR IF YES
2295	011242	012767	011206	167532		MOV	#6\$,LAD		;SETUP LOOP ADDRESS
2296	011250	104000				SCOPE			
2297	011252	012700	040000		7\$:	MOV	#40000,R0		;START OF BUFFER
2298	011256	016701	014360			MOV	WORDCT,R1		;BUFFER SIZE
2299	011262	010002			9\$:	MOV	R0,R2		
2300	011264	022002				CMP	(R0)+,R2		;DATA GOOD?
2301	011266	001004				BNE	8\$		;BR IF DATA NO GOOD
2302	011270	005722				TST	(R2)+		
2303	011272	005301				DEC	R1		;DONE?
2304	011274	001372				BNE	9\$		;BR IF NO
2305	011276	000413				BR	10\$		;CONTINUE THE TST
2306	011300	010267	006452		8\$:	MOV	R2,EXPS		
2307	011304	014067	006450			MOV	-(R0),RECS		
2308	011310	104405				HLT	+5		;DATA COMPARE ERROR
2309	011312	012767	011262	167462		MOV	#9\$,LAD		;SETUP LOOP ADDRESS
2310	011320	104000				SCOPE			
2311	011322	005720				TST	(R0)+		;TEST REST OF BUFFER
2312	011324	000756				BR	9\$		;TEST MORE
2313	011326	032767	001000	014276	10\$:	BIT	#B9,FLAG		;LAST PAGE?
2314	011334	001657				BEQ	1\$		;BR IF NO
2315						;CHECK THE SETTING OF NON EXISTENT MEMORY FLAG IN RPER.			
2316									
2317	011336	032767	000100	014266	NONEX:	BIT	#B6,FLAG		;IS MEMORY MANAGEMENT ON?
2318	011344	001002				BNE	4\$		;BR IF NOT ON
2319	011346	005037	177572			CLR	#SR0		;IF ON TURN OFF
2320	011352	012767	000001	014262	4\$:	MOV	#1,WORDCT		;SETUP WORD COUNT
2321	011360	012767	160000	014266		MOV	#160000,BUF		;LOAD ADDR TO FORCE NON EX MEMORY
2322	011366	004567	003542			JSR	R5,FUNCT		;ISSUE A WRITE COMMAND AND
2323	011372	000063			.WORD	63			;EXPECT NON EX MEMORY
2324	011374	105777	014206			TSTB	DRPCS		;WAIT FOR READY
2325	011400	100375				BPL	-4		
2326	011402	032777	000004	014214		BIT	#B2,DRPER		;DID NON EX MEMORY SET
2327	011410	001002				BNE	1\$		;BRANCH IF SET
2328	011412	104400				HLT			;NON EX MEMORY DID NOT SET
2329	011414	000413				BR	2\$		
2330	011416	032777	040000	014162	1\$:	BIT	#B14,DRPCS		;DID HARD ERROR SET ON NON EX MEMORY
2331	011424	001002				BNE	3\$		;BRANCH IF SET
2332	011426	104400				HLT			;HARD ERROR DID NOT SET
2333	011430	000405				BR	2\$		
2334	011432	032777	100000	014146	3\$:	BIT	#B15,DRPCS		;DID THE ERROR FLAG SET AFTER NON EX MEMORY
2335	011440	001001				BNE	2\$		;BRANCH IF SET
2336	011442	104400				HLT			;ERROR FLAG DID NOT SET
2337	011444	012767	011352	167330	2\$:	MOV	#4\$,LAD		
2338	011452	104000				SCOPE			
2339									
2340						;IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS			
2341						;BITS.			
2342									
2343	011454	012767	012252	166322	EXTTST:	MOV	#EXTTRP,4		;SETUP TIMEOUT TRAP
2344	011462	012767	000340	166316		MOV	#PRI7,6		
2345	011470	005737	177572			TST	#SR0		;IF MEMORY MANAGEMENT IS NOT
2346						;AVAILABLE THE PROGRAM WILL TRAP			
2347						;AND TRANSFER TO END OF THE TEST			

2348	011474	012767	012236	166302		MOV	#EXTRP,4	
2349	011502	012737	007600	172356		MOV	#7600,2#KIPAR7	; OPEN I/O REGISTERS
2350	011510	005037	172340			CLR	2#KIPAR0	; FREE FIRST 4K
2351	011514	012737	000200	172342		MOV	#200,2#KIPAR1	; ENABLE SECOND 4K
2352	011522	012737	002000	172344		MOV	#2000,2#KIPAR2	
2353	011530	012737	177406	172300		MOV	#400*256.-400+UP+RW,2#KIPDR0	; SET KIPDR0=RW UP 400 BLOCKS
2354	011536	012737	177406	172302		MOV	#400*256.-400+UP+RW,2#KIPDR1	; SET KIPDR1=RW UP 400 BLOCKS
2355	011544	012737	177406	172304		MOV	#400*256.-400+UP+RW,2#KIPDR2	; SET KIPDR2=RW UP 400 BLOCKS
2356	011552	012737	177406	172316		MOV	#400*256.-400+UP+RW,2#KIPDR7	; SET KIPDR7=RW UP 400 BLOCKS
2357	011560	012737	000001	177572		MOV	#1,2#SRO	; TURN ON MEMORY MANAGEMENT
2358	011566	012702	040000			MOV	#40000,R2	; R2 EQUALS BASE ADDR
2359	011572	012712	177777		7S:	MOV	#177777,(R2)	; INSERT PATTERN INTO 200000
2360	011576	012767	000002	014036		MOV	#2,WORDC↑	; SETUP WORDCOUNT
2361	011604	005067	010142			CLR	MEX	; CLEAR EXTENDED MEMORY BITS FOR CONTROLER
2362	011610	012767	177777	014036		MOV	#177777,BUF	; SETUP BUS ADDR
2363	011616	004567	003312			JSR	R5,FUNCT	; WRITE TWO WORDS ON DISK. RPBA
2364	011622	000003			.WORD	3		; STARTS AT 177777 TO FORCE CARRY
2365								; TO SET MEX0
2366	011624	105777	013756			TSTB	2RPCS	; WAIT FOR READY
2367	011630	100375				BPL	.-4	
2368	011632	005777	013750			TST	2RPCS	
2369	011636	100002				BPL	1S	
2370	011640	104400				HLT		; STATUS ERROR AFTER 2 WORD WRITE
2371	011642	000447				BR	2S	; USING MEX0
2372	011644	032777	000020	013734	1S:	BIT	#84,2RPCS	; MEX0 SHOULD HAVE SET?
2373	011652	001002				BNE	3S	; BRANCH IF SET
2374	011654	104400				HLT		; MEX0 DID NOT SET
2375	011656	000441				BR	2S	
2376	011660	005012			3S:	CLR	(R2)	; CLEAR LOCATION 200000
2377	011662	004567	003246			JSR	R5,FUNCT	; READ TWO WORDS INTO LOCATIONS
2378	011666	000005			.WORD	5		; 177777 AND 200000.
2379	011670	105777	013712			TSTB	2RPCS	; WAIT FOR READY
2380	011674	100375				BPL	.-4	
2381	011676	005777	013704			TST	2RPCS	; ANY ERRORS?
2382	011702	100002				BPL	4S	; BRANCH IF NO
2383	011704	104400				HLT		; ERROR AFTER READING 2 WORDS
2384	011706	000425				BR	2S	
2385	011710	032777	000020	013670	4S:	BIT	#84,2RPCS	; DID MEX0 SET?
2386	011716	001002				BNE	5S	; BRANCH IF YES
2387	011720	104400				HLT		; MEX0 DID NOT SET AFTER 2 WORD
2388	011722	000417				BR	2S	; READ STARTING AT 177777
2389	011724	022712	177777		5S:	CMP	#177777,(R2)	; WAS DATA READ INTO LOCATION
2390	011730	001407				BEQ	6S	; 200000 CORRECTLY? - BRANCH IF YES
2391	011732	012767	177777	006016		MOV	#177777,EXPS	
2392	011740	011267	006014			MOV	(R2),RECS	
2393	011744	104401				HLT	+1	; DATA COMPARE ERROR AT 200000
2394	011746	000405				BR	2S	; IF RECEIVED=0 - LOCATION WASN'T ACCESSED
2395	011750	032777	000040	013630	6S:	BIT	#85,2RPCS	; MEX1 SHOULD BE CLEAR
2396	011756	001401				BEQ	2S	; BRANCH IF CLEAR
2397	011760	104400				HLT		; MEX1 IS SET - SHOULD NOT BE
2398	011762	012767	011572	167012	2S:	MOV	#7S,LAD	; SETUP ERROR LOOP
2399	011770	104000				SCOPE		
2400	011772	012737	004000	172344	EXTT1:	MOV	#4000,2#KIPAR2	
2401	012000	012702	040000			MOV	#40000,R2	; R2 EQUALS THE BASE ADDR
2402	012004	012712	177777		7S:	MOV	#177777,(R2)	; INSERT PATTERN INTO 400000
2403	012010	012767	177777	013636		MOV	#177777,BUF	; SETUP BUS ADDR

2404	012016	004567	003112			JSR	R5,FUNCT	
2405	012022	000023		.WORD	23	TSTB	2RPCS	;WAIT FOR READY
2406	012024	105777	013556			BPL	-4	
2407	012030	100375				TST	2RPCS	;ANY ERRORS?
2408	012032	005777	013550			BPL	45	;BRANCH IF NO
2409	012036	100002				HLT		
2410	012040	104400				BR	25	;ERROR AFTER READING 2 WORDS
2411	012042	000455				BIT	#84,2RPCS	;DID MEX0 CLEAR?
2412	012044	032777	000020	013534	45:	BEQ	55	;BRANCH IF YES
2413	012052	001402				HLT		;MEX0 DID NOT CLEAR AFTER 2 WORD
2414	012054	104400				BR	25	;READ STARTING AT 377777
2415	012056	000447				BIT	#85,2RPCS	;DID MEX1 SET?
2416	012060	032777	000040	013520	55:	BNE	105	;BRANCH IF YES
2417	012066	001002				HLT		;MEX1 DI NOT SET WITH A TWO WORD
2418	012070	104400						;TRANSFER STARTING AT 377777
2419								
2420	012072	000441				BR	25	
2421	012074	005012			105:	CLR	(R2)	;CLEAR LOCATION 400000
2422	012076	004567	003032			JSR	R5,FUNCT	;READ TWO WORDS STARTING AT 377777
2423	012102	000025		.WORD	25	TSTB	2RPCS	;WAIT FOR READY
2424	012104	105777	013476			BPL	-4	
2425	012110	100375				TST	2RPCS	;ANY ERRORS?
2426	012112	005777	013470			BPL	115	;BRANCH IF NO
2427	012116	100002				HLT		;ERROR WHILE READING TWO WORDS
2428	012120	104400				BR	25	
2429	012122	000425				BIT	#84,2RPCS	;DID MEX0 CLEAR?
2430	012124	032777	000020	013454	115:	BEQ	125	;BRANCH IF YES
2431	012132	001402				HLT		;MEX0 DID NOT CLEAR AFTER 2 WORD
2432	012134	104400				BR	25	;READ STARTING AT 377777
2433	012136	000417				CMP	#177777,(R2)	;WAS DATA READ INTO LOCATION 400000
2434	012140	022712	177777		125:	BEQ	65	;CORRECTLY? - BRANCH IF YES
2435	012144	001407				MOV	#177777,EXPS	
2436	012146	012767	177777	005602		MOV	(R2),RECS	
2437	012154	011267	005600			HLT	+1	;DATA COMPARE ERROR AT 400000 IF
2438	012160	104401				BR	25	;RECEIVED=0 - LOCATION WASN'T ACCESSED
2439	012162	000405				BIT	#85,2RPCS	;DID MEX1 SET?
2440	012164	032777	000040	013414	65:	BNE	25	
2441	012172	001001				HLT		;MEX1 DID NOT SET AFTER 2 WORD TRANSFER
2442	012174	104400						;STARTING AT 377777
2443								;SETUP ERROR LOOP
2444	012176	012767	012004	166576	25:	MOV	#75,LAD	
2445	012204	104000				SCOPE		
2446	012206	000413				BR	EXTRP	;CLEAR MEMORY MANAGEMENT
2447	012210	005267	013452		EXTEND:	INC	PASSC	;INCREMENT ITERATION COUNT
2448	012214	001402				BEQ	15	
2449	012216	000167	176362			JMP	RMENT	
2450	012222	032737	002000	177570	15:	BIT	#810,2#SWR	;LOOP ON TEST?
2451	012230	001420				BEQ	DATAT	
2452	012232	000167	176346			JMP	RMENT	
2453								
2454	012236	032767	000100	013366	EXTRP:	BIT	#86,FLAG	;USING MEMORY MANAGEMENT?
2455	012244	001402				BEQ	EXTRP	;BR IF YES
2456	012246	005037	177572			CLR	#SRO	;NO TURN IT OFF
2457	012252	012706	000500		EXTTRP:	MOV	#STKPTR,SP	;RESTORE STACK
2458	012256	012767	000006	165520		MOV	#6,4	
2459	012264	005067	165516			CLR	6	

E05

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:19 PAGE 56  
DZRPB.P11 \*\*\*\*\* TEST 4 \*\*\*\*\*

2460 012270 000747

BR EXTEND

.  
.  
.

.SBTTL \*\*\*\*\* TEST 5 \*\*\*\*\*

;WRITE, WRITE CHECK, AND READ OPERATIONS ARE PERFORMED ON THE DRIVE  
 ;THE DATA IS FIRST WRITTEN AND THEN WRITE CHECKED. THEN THE DATA  
 ;IS READ. IF THE DATA IS TO BE COMPARED, THE INPUT BUFFER IS CLEARED  
 ;RIGHT AFTER READ IS ISSUED. THEN THE DATA IS COMPARED WHILE READ  
 ;IS IN PROGRESS. THIS IS DONE TO IMPROVE EFFICIENCY. THIS SEQUENCE  
 ;IS REPEATED FOR THE ENTIRE PACK SURFACE FOR EACH OF THE 22 PATTERNS.  
 ;ERRORS OCCURING USING RANDOM SEEKS DURING THE WRITE CHECK OR ARE  
 ;RECOVERABLE AFTER FIRST RETRY DURING READ OR ARE ELIMINATED BY  
 ;DISABLING RANDOM SEEKS ARE PROBABLY DUE TO VIBRATING HEADS  
 ;AFTER A SEEK.

2461									
2462									
2463									
2464									
2465									
2466									
2467									
2468									
2469									
2470									
2471									
2472									
2473									
2474	012272	012767	000005	013376		DATAT:	MOV	#5,TESTNO	
2475	012300	004567	166760				JSR	RS,PRINT\$	;PRINT MESSAGE
2476	012304	023740					MES6		
2477	012306	016767	013364	167246			MOV	TESTNO,TTY	
2478	012314	004767	167036				JSR	PC,PRINT\$	;TYPE LOCATION-SUPPRESS ZEROS
2479	012320	005067	013376				CLR	CNTA	;INITIALIZE READ COUNTER
2480	012324	005067	007422				CLR	MEX	;CLEAR EXTENDED MEMORY BITS IN CONTROLER
2481	012330	016700	013316			RDATAT:	MOV	PATNU,RO	;GET PATTERN NO.
2482	012334	000241					CLC		
2483	012336	006000					ROR	RO	
2484	012340	010037	177570				MOV	RO,#SWR	;DISPLAY PATTERN NO. IN USE
2485	012344	005067	013274				CLR	CYLINDER	
2486	012350	005067	013272				CLR	DMA	
2487	012354	012737	000200	177776			MOV	#PRI4,#PSW	;ENABLE INT SYSTEM
2488	012362	016767	013270	013252			MOV	SWRDCI,WRDCT	
2489	012370	012777	015224	013204			MOV	#DKINT,#VECTOR	;SETUP DISK VECOTR
2490	012376	012777	000340	013200			MOV	#340,#STATUS	
2491	012404	032767	000500	013220			BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM. MANAG
2492	012412	001006					BNE	DATP	;BR IF TRUE
2493	012414	004767	007344				JSR	PC,PARINT	;INITIALIZE KIPARS
2494	012420	000240					NOP		
2495	012422	000240					NOP		
2496	012424	004767	007454				JSR	PC,PARREG	;NEEDED FOR CONFORMITY
2497	012430	004767	004404			DATP:	JSR	PC,PASEL	;SETUP "MEX", "WRDCT", "BUF"
2498	012434	032767	000500	013170			BIT	#500,FLAG	;GENERATE PATTERN
2499	012442	001403					BEQ	1\$	;BK MEM. MANAG. OR NO MEM. MANAG?
2500	012444	012767	025730	013202			MOV	#OUTBUF,BUF	;BR IF NOT TRUE
2501	012452	032767	040000	013152		1\$:	BIT	#B14,FLAG	;SETUP BUFFER ADDR
2502	012460	001426					BEQ	WRICK	;WRITE?
2503	012462	004767	003434			LDAT:	JSR	PC,OPDSEL	;BRANCH IF NO
2504	012466	004767	003126				JSR	PC,SEEK	;ANY OPERATOR ADDR PARAMETERS?
2505	012472	004567	002436				JSR	RS,FUNCT	;GO DO A RANDOM SEEK
2506	012476	000103				.WORD	103		;WRITE WITH INTERRUPTS
2507	012500	032737	010000	177570			BIT	#B12,#SWR	;DETERMINE HOW TO WAIT FOR INT
2508	012506	001003					BNE	1\$	
2509	012510	004767	007534				JSR	PC,NPR	;GENERATE WORSE CASE NPR CYCLES
2510	012514	000401					BR	2\$	
2511	012516	000001				1\$:	WAIT		
2512	012520	012767	012462	166254		2\$:	MOV	#LDAT,LDAT	;SETUP LOOP ADDR
2513	012526	104000					SCOPE		
2514	012530	004767	003424				JSR	PC,DISBUF	;PREPARE NEW DISK ADDR
2515	012534	000752					BR	LDAT	
2516	012536	032767	020000	013066		WRICK:	BIT	#B13,FLAG	;WRITE CHECK?

2517	012544	001424				BEQ	DREAD		;BRANCH IF NO
2518	012546	004767	003350		3\$:	JSR	PC,OPDSEL		;ANY OPERATOR ADDR PARAMETERS?
2519	012552	004567	002356			JSR	RS,FUNCT		;WRITE CHECK THE DATA
2520	012556	000107			.WORD	107			
2521	012560	032737	010000	177570		BIT	#B12,2#SWR		;DETERMINE HOW TO WAIT FOR INT
2522	012566	001003				BNE	1\$		
2523	012570	004767	007454			JSR	PC,NPR		;GENERATE WORSE CASE NPR CYCLES
2524	012574	000401				BR	2\$		
2525	012576	000001			1\$:	WAIT			
2526	012600	012767	012546	166174	2\$:	MOV	#3\$,LAD		;SETUP LOOP ADDR
2527	012606	104000				SCOPE			
2528	012610	004767	003344			JSR	PC,DISBUF		;PREPARE NEW DISK ADDR
2529	012614	000754				BR	3\$		
2530	012616	032767	010000	013006	DREAD:	BIT	#B12,FLAG		;READ?
2531	012624	001002				BNE	1\$		;BRANCH IF YES
2532	012626	000167	000444			JMP	MSTR		;GET AROUND READ
2533	012632	032767	000500	012772	1\$:	BIT	#500,FLAG		;BK WITH MEM MANAG. OR NO MEM MANAG.
2534	012640	001006				BNE	READ1		;BR IF TRUE
2535	012642	004767	007116			JSR	PC,PARINT		;INITIALIZE KIPARS NEEDED
2536	012646	000240				NOP			;FOR CONFORMITY
2537	012650	000240				NOP			
2538	012652	004767	007226			JSR	PC,PARREG		;SETUP "BUF" WRDCT" "MEX"
2539	012656	005067	013040		READ1:	CLR	CNTA		;INITIALIZE READ COUNTER
2540	012662	005067	013004		ESH:	CLR	RDERR		;CLEAR READ ERROR COUNT
2541	012666	004767	003230			JSR	PC,OPDSEL		;ANY OPERATOR ADDR PARAMETERS?
2542	012672	004767	002722			JSR	PC,SEEK		;GO DO A RANDOM SEEK
2543	012676	005067	171576		DSKRD:	CLR	INTFLG		;CLEAR THE INTERRUPT FLAG
2544	012702	004567	002226			JSR	RS,FUNCT		;READ THE DATA
2545	012706	000105			.WORD	105			
2546	012710	032737	001000	177570		BIT	#B9,2#SWR		;COMPARE DATA?
2547	012716	001411				BEQ	1\$		;BRANCH IF YES
2548	012720	032737	010000	177570		BIT	#B12,2#SWR		;DETERMINE HOW TO WAIT FOR INT
2549	012726	001003				BNE	2\$		
2550	012730	004767	007314			JSR	PC,NPR		;GENERATE WORSE CASE NPR CYCLES
2551	012734	000447				BR	3\$		
2552	012736	000001			2\$:	WAIT			
2553	012740	000445				BR	3\$		
2554	012742	016700	012674		1\$:	MOV	WRDCT,RO		;CLEAR THE INPUT BUFFER
2555	012746	012702	000020			MOV	#16.,R2		
2556	012752	012701	025730			MOV	#OUTBUF,R1		
2557	012756	032767	000500	012646		BIT	#500,FLAG		;BK WITH MEM. MANAG. OR NO MEM. MANAG.
2558	012764	001002				BNE	4\$		;BR IF TRUE
2559	012766	012701	040000			MOV	#40000,R1		;START OF MEM. MANAG. OUT BUFF
2560	012772	005700			4\$:	TST	RO		;TEST WORD COUNT IS IT AT ZERO?
2561	012774	100422				BMI	10\$		;BR IF CLEAR IS FINISHED
2562	012776	005021				CLR	(R1)+		
2563	013000	005021				CLR	(R1)+		
2564	013002	005021				CLR	(R1)+		
2565	013004	005021				CLR	(R1)+		
2566	013006	005021				CLR	(R1)+		
2567	013010	005021				CLR	(R1)+		
2568	013012	005021				CLR	(R1)+		
2569	013014	005021				CLR	(R1)+		
2570	013016	005021				CLR	(R1)+		
2571	013020	005021				CLR	(R1)+		
2572	013022	005021				CLR	(R1)+		

2573	013024	005021			CLR	(R1)+	
2574	013026	005021			CLR	(R1)+	
2575	013030	005021			CLR	(R1)+	
2576	013032	005021			CLR	(R1)+	
2577	013034	005021			CLR	(R1)+	
2578	013036	160200			SUB	R2,R0	;DECREMENT WORD COUNT
2579	013040	001354			BNE	4\$	
2580	013042	004767	004750	10\$:	JSR	PC,COMPAR	;COMPARE THE DATA
2581	013046	105777	012534		TSTB	DRPCS	;WAIT FOR READY
2582	013052	100375			BPL	-4	
2583	013054	005767	166160	3\$:	TST	ERRFLG	;WERE THERE ANY ERRORS
2584	013050	001424			BEQ	5\$	;BRANCH IF NO
2585	013062	005267	012604		INC	RDERR	;UPDATE ERROR COUNT
2586	013066	022767	000024	012576	CMP	#20.,RDERR	;MORE THAN 20 ERRORS?
2587	013074	001416			BEQ	5\$	;BRANCH IF YES
2588	013076	022767	000012	012566	CMP	#10.,RDERR	;IS THIS TENTH ERROR?
2589	013104	001274			BNE	DSKRD	;BRANCH IF NO
2590	013106	112777	000015	012472	MOVB	#15,DRPCS	;HOME THE HEADS
2591	013114	105777	012466		TSTB	DRPCS	;WAIT FOR DONE
2592	013120	100375			BPL	-4	
2593	013122	005777	012500		TST	DRPDS	
2594	013126	100375			BPL	-4	;WAIT FOR READY
2595	013130	000662			BR	DSKRD	
2596	013132	005767	012534	5\$:	TST	RDERR	
2597	013136	001417			BEQ	6\$	
2598	013140	022767	000013	012524	CMP	#11.,RDERR	;WAS READ ERROR STILL THERE AFTER A RECALL
2599	013146	002003			BGE	9\$	;BR IF NO
2600	013150	012767	000002	012544	MOV	#2,CNTA	;YES DON'T TRY TO READ AGAIN UNLESS LOOPING
2601	013156			9\$:			
2602	013156	004567	166102		JSR	R5,PRINT\$	;PRINT MESSAGE
2603	013162	023753			MES7		
2604	013164	016767	012502	166370	MOV	RDERR,TTY	
2605	013172	004767	166160		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2606	013176	005067	012470	6\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
2607	013202	012767	012662	165572	MOV	#ESH,LAD	;LOOP ADDR
2608	013210	104000			SCOPE		
2609	013212	005267	012504		INC	CNTA	;INCREMENT READ COUNTER
2610	013216	022767	000003	012476	CMP	#3,CNTA	;DONE 3 READS?
2611	013224	001216			BNE	ESH	;BR IF NO
2612	013226	032767	000500	012376	BIT	#500,FLAG	;BK MEM. MANAG. OR NO MEM. MANAG.
2613	013234	001012			BNE	8\$	;BR IF TRUE
2614	013236	032767	001000	012366	BIT	#B9,FLAG	;LAST PAGE TRANSFERED?
2615	013244	001402			BEQ	7\$	;BR IF NO
2616	013246	004767	006512		JSR	PC,PARINT	;GO REINITIALIZE KIPARS
2617	013252	004767	006576	7\$:	JSR	PC,PARINC	;GO INCREMENT KIPARS BY 1K
2618	013256	004767	006622		JSR	PC,PARREG	;GO SETUP "MEX" "WRDCT" "BUF"
2619	013262	004767	002672	8\$:	JSR	PC,DISBUF	;GET NEW DISK ADDR
2620	013266	000401			BR	11\$	
2621	013270	000402			BR	MSTR	
2622	013272	000167	177360	11\$:	JMP	READ1	
2623	013276	032767	000040	012326	MSTR:	BIT	#B5,FLAG
2624	013304	001402			BEQ	1\$	;LOOPING ON AN OPERATOR ADDR?
2625	013306	000167	177116		JMP	DATP	;NO CONTINUE
2626	013312	005767	012314	1\$:	TST	FLAG	;YES
2627	013316	100002			BPL	2\$	
2628	013320	000167	001034		JMP	MULCHK	;UNDER PROGRAM CONTROL ;OPERATOR SELECTED PATTERN

2629	013324	062767	000002	012320	2S:	AUD	#2,PATNU	;INC PATTERN INDEX
2630	013332	022767	000036	012312		CMP	#36,PATNU	;PATTERNS EXCEEDED?
2631	013340	001402				BEQ	3S	
2632	013342	000167	176762			JMP	RDATAT	;NOT YET
2633	013346	005067	012300		3S:	CLR	PATNU	;LAST PATTERN USED
2634	013352	032737	002000	177570		BIT	#B10,2#SWR	;LOOP ON TEST?
2635	013350	001402				BEQ	RANEX	;NO..GO TO RANDOM TEST
2636	013362	000167	176742			JMP	RDATAT	;YES



```

2637 .SBTTL ***** TEST 6 *****
2638
2639 ;THIS IS A RANDOM ADDRESS AND DATA TEST.
2640 ;ERRORS OCCURING WHICH ARE RECOVERABLE AFTER THE FIRST RETRY ARE PROBABLY
2641 ;DUE TO VIBRATING HEADS AFTER A SEEK, TRY DISABLEING RANDOM SEEKS.
2642
2643 013366 012767 000006 012302 RANEX: MOV #6,TESTNO
2644 013374 004567 165664 JSR RS,PRINTS ;PRINT MESSAGE
2645 013400 023740 MES6
2646 013402 016767 012270 166152 MOV TESTNO,TTY
2647 013410 004767 165742 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2648 013414 005067 006332 CLR MEX ;CLEAR EXTENDED MEMORY BITS IN CONTROLER
2649 013420 004767 003216 RRANEX: JSR PC,INIT
2650 013424 032767 000500 012200 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.
2651 013432 001004 BNE 1$ ;BR IF TRUE
2652 013434 004767 006324 JSR PC,PARINT ;INITALIZE ALL KIPARS
2653 013440 000240 NOP ;NEEDED FOR CONFORMITY
2654 013442 000240 NOP
2655 013444 012767 000034 012200 1$: MOV #34,PATNU
2656 013452 012767 175300 012206 MOV #-2500,PASSC ;SET UP PASS COUNT
2657 013460 012737 000200 177776 MOV #PRI4,#PSW
2658 013466 012767 000700 012146 WRLG: MOV #700,WRDCT ;SET UP WORD COUNT TO 1+ SECTOR
2659 013474 016767 012142 012200 MOV WRDCT,WORK
2660 013502 012701 025730 MOV #OUTBUF,R1
2661 013506 032767 000500 012116 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
2662 013514 001002 BNE 4$ ;BR IF TRUE
2663 013516 012701 040000 MOV #40000,R1 ;START OF OUTBUFFER WITH MEMORY MANAG.
2664 013522 004767 003420 4$: JSR PC,RANDOM ;GENERATE RANDOM PATTERN
2665 013526 1$:
2666 013526 004767 166032 JSR PC,RANDS ;GENERATE TWO RANDOM NOS.
2667 013532 016767 166144 012142 MOV LONUM,WORK
2668 013540 016767 166134 012136 MOV HINUM,WORK1
2669 013546 042767 177000 012126 BIC #177000,WORK
2670 013554 022767 000625 012120 CMP #625,WORK ;FORM RANDOM CYL ADDR
2671 013562 002761 BLT 1$
2672 013564 016767 012112 012052 MOV WORK,CYLINDER ;SAVE IT
2673 013572 042767 160360 012104 BIC #160360,WORK1
2674 013600 122767 000010 012076 CMPB #10,WORK1 ;FORM RANDOM SECTOR ADDR
2675 013606 101003 BHI 2$
2676 013610 042767 000010 012066 BIC #10,WORK1
2677 013616 122767 000023 012061 2$: CMPB #23,WORK1+1
2678 013624 101003 BHI 3$ ;FORM RANDOM HEAD ADDR
2679 013626 142767 000014 012051 BICB #14,WORK1+1
2680 013634 016767 012044 012004 3$: MOV WORK1,DMA ;SAVE DESK ADDR.
2681 013642 012767 025730 012004 RANLOP: MOV #OUTBUF,BUF ;SETUP OUTPUT BUFFER
2682 013650 032767 000500 011754 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
2683 013656 001003 BNE 13$ ;BR IF TRUE
2684 013660 012767 040000 011766 MOV #40000,BUF ;START OF OUTBUFFER WITH MEMORY MANAG.
2685 013666 004767 001726 13$: JSR PC,SEEK ;DO A RANDOM SEEK
2686 013672 004567 001236 JSR RS,FUNCT ;WRITE RANDOM DATA AND
2687 013676 000103 .WORD 103 ;ENABLE INTERRUPTS
2688 013700 032737 010000 177570 BIT #812,#SWR ;DETERMINE HOW TO WAIT FOR INT
2689 013706 001003 BNE 2$
2690 013710 004767 006334 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
2691 013714 000401 BR 4$
2692 013716 000001 2$: WAIT

```

2693	013720	012767	013642	165054	4\$:	MOV	#RANLOP,LAD	;SETUP LOOP ADDR
2694	013726	104000				SCOPE		
2695	013730	004567	001200		7\$:	JSR	RS,FUNCT	;WRITE CHECK THE DATA AND
2696	013734	000107			.WORD	107		;ENABLE INTERRUPT
2697	013736	032737	010000	177570		BIT	#B12,2#SWR	;HOW TO WAIT FOR INT?
2698	013744	001003				BNE	1\$	
2699	013746	004767	006276			JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2700	013752	000401				BR	5\$	
2701	013754	000001			1\$:	WAIT		
2702	013756	012767	013730	165016	5\$:	MOV	#7\$,LAD	;SETUP LOOP ADDR
2703	013764	104000				SCOPE		
2704	013766	005067	011730			CLR	CNTA	;INITALIZE READ COUNTER
2705	013772	005067	011674		8\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
2706	013776	004767	001616			JSR	PC,SEEK	;DO A RANDOM SEEK
2707	014002	005067	170472		11\$:	CLR	INTFLG	
2708	014006	004567	001122			JSR	RS,FUNCT	;READ RANDOM DATA AND
2709	014012	000105			.WORD	105		;ENABLE INTERRUPT
2710	014014	012700	000700			MOV	#700,R0	;SET UP TO CLEAR INPUT BUFFER
2711	014020	012702	000020			MOV	#16,R2	
2712	014024	012701	025730			MOV	#OUTBUF,R1	;START OF INPUT BUFFER
2713	014030	032767	000500	011574		BIT	#500,FLAG	;BK WITH MEM. MANAG. OR NO MEM MANAG?
2714	014036	001002				BNE	15\$	;BR IF TRUE
2715	014040	012701	040000			MOV	#40000,R1	;START OF OUTBUF WITH MEM. MANAG.
2716	014044				15\$:			
2717	014044	005021				CLR	(R1)+	
2718	014046	005021				CLR	(R1)+	
2719	014050	005021				CLR	(R1)+	
2720	014052	005021				CLR	(R1)+	
2721	014054	005021				CLR	(R1)+	
2722	014056	005021				CLR	(R1)+	
2723	014060	005021				CLR	(R1)+	
2724	014062	005021				CLR	(R1)+	
2725	014064	005021				CLR	(R1)+	
2726	014066	005021				CLR	(R1)+	
2727	014070	005021				CLR	(R1)+	
2728	014072	005021				CLR	(R1)+	
2729	014074	005021				CLR	(R1)+	
2730	014076	005021				CLR	(R1)+	
2731	014100	005021				CLR	(R1)+	
2732	014102	005021				CLR	(R1)+	
2733	014104	160200				SUB	R2,R0	;DECREMENT THE WORD COUNT
2734	014106	001356				BNE	15\$	;GO CLR MORE
2735	014110	032737	010000	177570		BIT	#B12,2#SWR	;HOW TO WAIT FOR INT?
2736	014116	001003				BNE	3\$	
2737	014120	004767	006124			JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2738	014124	000401				BR	6\$	
2739	014126	000001			3\$:	WAIT		
2740	014130	032737	001000	177570	6\$:	BIT	#B9,2#SWR	;COMPARE FOR ERRORS?
2741	014136	001006				BNE	9\$	;BRANCH IF NO
2742	014140	032777	040000	011440		BIT	#B14,2RPCS	;HARD ERROR?
2743	014146	001002				BNE	9\$	;BRANCH IF YES
2744	014150	004767	003642			JSR	PC,COMPARE	;COMPARE DATA FOR ERRORS
2745	014154	005767	165060		9\$:	TST	ERRFLG	;READ ERROR?
2746	014160	001424				BEQ	10\$	;BRANCH IF NO
2747	014162	005267	011504			INC	RDERR	;UPDATE ERROR COUNT
2748	014166	022767	000024	011476		CMP	#20.,RDERR	;20 ERRORS YET?

2749	014174	001416			BEQ	10\$		;BRANCH IF YES
2750	014176	022767	000012	011466	CMP	#10.,RDERR		;IS THIS TENTH ERROR?
2751	014204	001276			BNE	11\$		;BRANCH IF NO
2752	014206	112777	000015	011372	MOVB	#15,DRPCS		;ISSUE HOME COMMAND
2753	014214	105777	011366		TSTB	DRPCS		;WAIT FOR DONE
2754	014220	100375			BPL	.-4		
2755	014222	005777	011400		TST	DRPOS		;WAIT FOR READT
2756	014226	100375			BPL	.-4		
2757	014230	000664			BR	11\$		
2758	014232	005767	011434		10\$: TST	RDERR		
2759	014236	001417			BEQ	21\$		
2760	014240	022767	000013	011424	CMP	#11.,RDERR		;DID RECALL HELP RECOVERY
2761	014246	002003			BGE	14\$		;BR IF YES
2762	014250	012767	000011	011444	MOV	#9.,CNTA		;NO, DON'T TRY READING AGAIN
2763	014256				14\$: JSR	R5,PRINT\$		;PRINT MESSAGE
2764	014256	004567	165002		MES7			
2765	014262	023753			MOV	RDERR,TTY		
2766	014264	016767	011402	165270	JSR	PC,PRINTS		;TYPE LOCATION-SUPRESS ZEROS
2767	014272	004767	165060		CLR	RDERR		;CLEAR READ ERROR COUNTER
2768	014276	005067	011370		MOV	#8\$,LAD		;SET UP LOOP ADDR
2769	014302	012767	013772	164472	21\$: SCOPE			
2770	014310	104000			INC	CNTA		;INCREMENT READ COUNTER
2771	014312	005267	011404		CMP	#10.,CNTA		;DONE 10 DISK READS?
2772	014316	022767	000012	011376	BNE	8\$		;BR IF NO
2773	014324	001222			INC	PASSC		;INCREMENT PASS COUNT
2774	014326	005267	011334		BEQ	12\$		;BRANCH IF DONE
2775	014332	001402			JMP	WRLG		;CONTINUE
2776	014334	000167	177126		12\$: CLR	PATNU		
2777	014340	005067	011306		BIT	#810,DRSWR		;LOOP ON TEST?
2778	014344	032737	002000	177570	BEQ	MULCHK		;NO
2779	014352	001402			JMP	RRANEX		;LOOP
2780	014354	000167	177040					
2781								
2782								
2783								
2784								
2785								
2786								
2787								
2788	014360	005067	011262		MULCHK: CLR	DMA		
2789	014364	005067	011254		CLR	CYLINDER		;CLEAR ADDRESS REGISTERS
2790	014370	032767	004000	011234	BIT	#811,FLAG		;ARE WE IN MULTI DISK MODE
2791	014376	001422			BEQ	REPOEN		;REPORT "END"
2792	014400	016767	011226	011274	MOV	FLAG,WORK		;WHAT DISK ARE WE ON
2793	014406	042767	177743	011266	BIC	#177743,WORK		;IF LAST DISK ON SYSTEM
2794	014414	026767	011262	011240	CMP	WORK,DSKNOR		;REPORT END
2795	014422	001004			BNE	INDRVE		
2796	014424	042767	000034	011200	BIC	#34,FLAG		
2797	014432	000404			BR	REPOEN		;REPORT "END" LAST DISK
2798	014434	062767	000004	011170	INDRVE: ADD	#4,FLAG		;INC. DISK NO.
2799	014442	000422			BR	EXTPP		;EXERCISE DISK
2800	014444	005267	164330		REPOEN: INC	ICNT		;INCREMENT PASS COUNTER
2801	014450	004567	164610		JSR	R5,PRINT\$		;PRINT MESSAGE
2802	014454	023550			MES1			;REPORT END OF PASS ,
2803	014456	016767	164316	165076	MOV	ICNT,TTY		
2804	014464	004767	164666		JSR	PC,PRINTS		;TYPE LOCATION-SUPRESS ZEROS

2805	014470	013701	000042	MOV	#42 R1	;GET MONITOR RETURN ADDRESS
2806	014474	001405		BEQ	EXTPP	;BRANCH IF NOT UNDER MONITOR
2807	014476	000005		RESET		
2808	014500	004711		MEXIT: JSR	PC, (R1)	;EXIT TO THE MONITOR
2809	014502	000240		NOP		
2810	014504	000240		NOP		
2811	014506	000240		NOP		
2812	014510	000167	167032	EXTPP: JMP	ADTST	;RECYCLE
2813						
2814						

2815  
 2816  
 2817  
 2818  
 2819  
 2820  
 2821  
 2822  
 2823  
 2824  
 2825  
 2826 014514 012706 000500  
 2827 014520 012767 015104 163276  
 2828 014526 012767 000340 163272  
 2829 014534 004567 002030  
 2830 014540 012777 000625 011050  
 2831 014546 052777 000011 011032  
 2832 014554 012700 025730  
 2833 014560 012720 025252  
 2834 014564 020067 005160  
 2835 014570 101773  
 2836 014572 012767 000625 011044  
 2837 014600 005067 011042  
 2838 014604 012767 000400 011030  
 2839 014612 012767 025730 011034  
 2840 014620 105777 010762  
 2841 014624 100375  
 2842 014626 005777 010774  
 2843 014632 100375  
 2844 014634 004567 000274  
 2845 014640 000003  
 2846 014642 105777 010740  
 2847 014646 100375  
 2848 014650 032777 100000 010730  
 2849 014656 001401  
 2850 014660 104400  
 2851 014662 012767 014514 164112  
 2852 014670 104000  
 2853 014672 004567 164366  
 2854 014676 023672  
 2855 014700 004567 000230  
 2856 014704 000005  
 2857 014706 105777 010674  
 2858 014712 100375  
 2859 014714 000771  
 2860  
 2861  
 2862  
 2863  
 2864 014716 012777 000001 010662  
 2865 014724 016767 010702 010750  
 2866 014732 000241  
 2867 014734 006067 010742  
 2868 014740 006067 010736  
 2869 014744 000367 010732  
 2870 014750 042767 174377 010724

.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.

PFTST: MOV #STKPTR, SP  
 MOV #PFD, 24 ;SET UP POWER FAIL VECTOR  
 MOV #PRI7, 26 ;LOCKOUT INTERRUPTS  
 JSR R5, DSKNOS ;SELECT THE UNIT  
 MOV #625, @RPCA ;SELECT CYLINDER 625  
 BIS #11, @RPCS ;ISSUE SEEK COMMAND  
 MOV #OUTBUF, RO  
 1\$: MOV #25252, (RO)+ ;FILL MEMORY WITH CHECKERBOARD  
 CMP RO, MEMSIZ ;PATTERN  
 BLOS 1\$  
 MOV #625, CYLINDER  
 CLR DMA  
 MOV #400, WRDCT  
 MOV #OUTBUF, BUF  
 TSTB @RPCS ;WAIT FOR DONE  
 BPL -4  
 TST @RPDS ;WAIT FOR UNIT READY  
 BPL -4  
 JSR R5, FUNCT ;WRITE 1 SECTOR OF CHECKERBOARD  
 .WORD 3  
 TSTB @RPCS ;WAIT FOR READY  
 BPL -4  
 BIT #B15, @RPCS ;ANY ERRORS?  
 BEQ 2\$  
 HLT ;DEVICE ERROR ON WRITE  
 2\$: MOV #PFTST, LAD  
 SCOPE  
 JSR R5, PRINT\$ ;PRINT MESSAGE  
 MESS ;HAVE POWER TURNED OFF  
 3\$: JSR R5, FUNCT ;GO INTO A LOOP READING  
 .WORD 5 ;THE DISK SURFACE  
 TSTB @RPCS  
 BPL -4  
 BR 3\$

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

PFT1: MOV #1, @RPCS ;CLEAR THE CONTROLLER  
 MOV FLAG, WORK ;GET UNIT NUMBER  
 CLC  
 ROR WORK  
 ROR WORK  
 SWAB WORK  
 BIC #174377, WORK



2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965

```

015134 004567 001430
015140 016777 010502 010452
015146 016777 010472 010442
015154 016777 010474 010432
015162 016777 010454 010422
015170 005477 010416
015174 011567 000022
015200 062705 000002
015204 056767 004542 000010
015212 056777 000004 010366
015220 000205
015222 000000

015224 005067 164010
015230 005777 010352
015234 100402
015236 000167 000316
015242 010667 163772
015246 005767 010420
015252 001404
015254 032737 000020 177570
015262 001522
015264 104400
015266 032777 000002 010312
015274 001012
015276 004567 163762
015302 024101
015304 016767 010362 164250
015312 004767 164040
015316 010667 163716

015322 017767 010270 000266
015330 017767 010264 000256
015336 042767 160360 000250
015344 032777 000001 010252
015352 001026
015354 032767 000017 000232
015362 001403
015364 005367 000224
015370 000417
015372 132767 000037 000215
015400 001406
015402 105367 000207
015406 052767 000011 000200
015414 000405
015416 012767 011411 000170
015424 005367 000166
015430
  
```

.SBTTL \*\*\* SUBROUTINES \*\*\*

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

```

FUNCT: JSR    R5,DSKNOS      ;SELECT THE UNIT
        MOV    DMA,DRPDA    ;SETUP DISK ADDR REG
        MOV    CYLINDER,DRPCA ;SETUP CYLINDER ADDR REG
        MOV    BUF,DRPBA    ;SETUP BUS ADDR REG
        MOV    WRDCT,DRPWC  ;SETUP WORD COUNT
        NEG   DRPWC        ;COMPLIMENT WORD COUNT
        MOV   (R5),FNCT    ;GET RPCS FUNCTION
        ADD   #2,R5        ;UPDATE RETURN ADDR
        BIS   MEX,FNCT     ;ADD EXTENDED MEMORY BITS
        BIS   FNCT,DRPCS   ;OUTPUT THE FUNCTION
        RTS   R5

FNCT:   0
  
```

;RP11 DISK INTERRUPT HANDLER

```

DKINT: CLR    ERRFLG      ;CLEAR THE ERROR FLAG
        TST   DRPCS      ;TEST FOR ERROR
        BMI   IS        ;JUMP IF NO ERRORS
        JMP   INTEXT    ;SET INTERRUPT ERROR FLAG
        MOV   SP,ERRFLG ;IS THIS THE FIRST ERROR ATTEMPT?
        TST   RDERR     ;BRANCH IF YES
        BEQ  Z$        ;TYPE ALL ERROR ATTEMPTS?
        BIT   #B4,DRSWR ;BRANCH IF NO
        BEQ  DKII     ;STATUS ERROR AFTER INTERRUPT
        HLT  Z$        ;CHECK FOR READ
        BIT   #B1,DRPCS ;BRANCH IF WRITTING
        BNE  DELMES   ;PRINT MESSAGE
        JSR  R5,PRINT$ ;GIVE # OF READ ATTEMPT
        JSR  RDERR,TTY ;TYPE LOCATION-SUPRESS ZEROS
        JSR  PC,PRINT$
        MOV  SP,ERRFLG

IS:     MOV   SP,ERRFLG
        TST   RDERR
        BEQ  Z$
        BIT   #B4,DRSWR
        BEQ  DKII
        BIT   #B1,DRPCS
        BNE  DELMES
        JSR  R5,PRINT$
        JSR  RDERR,TTY
        JSR  PC,PRINT$
        MOV  SP,ERRFLG

Z$:     HLT

DELMES: MOV   DRPCA,INT1 ;GET CYLINDER ADDR
        MOV   DRPDA,INT0 ;GET HEAD AND SECTOR ADDR
        BIC   #160360,INT0 ;CLEAR UNWANTED BITS
        BIT   #B0,DRPER ;WAS IT AN ADDR ERROR?
        BNE  REDAC     ;BRANCH IF YES
        BIT   #17,INT0 ;IS SECTOR = TO 0
        BEQ  DECTK    ;YES - BRANCH
        DEC  INTO     ;BACK UP COUNT
        BR   REDAC

DECTK: BITB  #37,INT0+1 ;IS HEAD = TO 0
        BEQ  DECCY    ;YES - BRANCH
        DECB INTO+1   ;BACK UP HEAD
        BIS  #11,INT0 ;SET UP SECTOR
        BR   REDAC

DECCY: MOV   #11411,INT0
        DEC  INT1

REDAC:
  
```

```

2966 015430 004567 163630 JSR R5,PRINTS ;PRINT MESSAGE
2967 015434 024115 MES14 ;REPORT CYLINDER ADDR
2968 015436 016767 000154 164116 MOV INT1,TTY
2969 015444 004767 163706 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2970 015450 005067 000142 CLR INT1
2971 015454 116767 000135 000134 MOVB INTO+1,INT1
2972 015462 004567 163576 JSR R5,PRINTS ;PRINT MESSAGE
2973 015466 024125 MES15 ;REPORT HEAD ADDR OF FAILURE
2974 015470 016767 000122 164064 MOV INT1,TTY
2975 015476 004767 163654 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2976 015502 116767 000106 000106 MOVB INTO,INT1
2977 015510 004567 163550 JSR R5,PRINTS ;PRINT MESSAGE
2978 015514 024136 MES16 ;REPORT SEC ADDR OF FAILURE
2979 015516 016767 000074 164036 MOV INT1,TTY
2980 015524 004767 163626 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2981 015530 032777 001000 010070 DK11: BIT #89,ARPC ;IS DRIVE UNSAFE?
2982 015536 001401 BEQ .+4
2983 015540 000000 HALT ;DRIVE UNSAFE
2984 015542 032777 002000 010056 BIT #810,ARPC ;SEEK INCOMPLETE?
2985 015550 001403 BEQ INTEXT ;BRANCH IF COMPLETE
2986 015552 112777 000015 010026 MOVB #15,ARPC ;RECALIBRATE
2987 015560 105777 010022 INTEXT: TST ARPC
2988 015564 100375 BPL .-4 ;WAIT FOR DONE
2989 015566 005777 010034 TST ARPC
2990 015572 100375 BPL .-4 ;WAIT FOR READY
2991 015574 005767 004676 TST BCKFLG ;DID WE COME FROM BACKGROUND TEST?
2992 015600 001402 BEQ IS ;BRANCH IF NO
2993 015602 012716 022464 MOV #NPRRET,(SP) ;MODIFY RETURN ADDR
2994 015606 010667 166666 IS: MOV SP,INTFLG ;SET INTERRUPT OCCURRED FLAG
2995 015612 000002 RTI
2996
2997 015614 000000 INTO: 0
2998 015616 000000 INT1: 0
2999
3000 ;THIS ROUTINE IF SWITCH 07 OFF ISSUES A SEEK RANDOM YET NOT DESTROYING
3001 ;THE CYLINDER UNDER TEST
3002
3003 015620 105737 177570 SEEK: TSTB @#SWR ;ISSUE A RANDOM SEEK?
3004 015624 100534 BMI SEEND ;BR IF NO
3005 015626 004567 000736 JSR R5,DSKNOS ;SELECT THE UNIT
3006 015632 005767 163402 TST ERRFLG ;ARE WE LOOPING ON ERROR?
3007 015636 001031 BNE 2$ ;BR IF YES-DO SAME SEEK AS THE ERROR
3008 015640 062767 123455 010052 IS: ADD #123455,SEEK1 ;GET A RANDOM NUMBER
3009 015646 042767 177000 010044 BIC #177000,SEEK1 ;CLEAR OUT UNWANTED CYLINDER BITS
3010 015654 022767 000625 010036 CMP #625,SEEK1 ;VALID CYLINDER NUMBER
3011 015662 002766 BLT IS ;BR IF TOO BIG
3012 015664 016746 010030 MOV SEEK1,-(SP) ;PUT NEW CYLINDER ON STACK
3013 015670 166716 007750 SUB CYLINDER,(SP) ;CREAT A DIFFERANCE ON STACK
3014 015674 100405 BMI 7$ ;BR IF DIFFERANCE IS -
3015 015676 021627 000200 CMP (SP),#200 ;IS DIFFERANCE BETWEEN CYLINDER AND RANDOM CYL
3016 ;GREATER THAN 200 CYLINDERS
3017 015702 100006 BPL 8$ ;YES, CLEANUP STACK AND CONTINUE
3018 015704 005726 9$: TST (SP)+ ;NO, CLEAN OFF STACK
3019 015706 000754 BR IS ;GET NEW CYLINDER RANDOM NUMBER
3020 015710 021627 177600 7$: CMP (SP),#-200
3021 015714 100401 BMI 8$ ;BR IF DIFFERANCE GRATER THAN 200 CLYINDERS

```



# E06

```

3022 015716 000772          BR      9$      ;DIFFERENCE LESS THAN 200 GET ANOTHER CYL NO
3023 015720 005726          TST     (SP)+   ;CLEANOFF STACK
3024 015722 017746 007654  2$:      MOV     @VECTOR,-(SP) ;SAVE VECTOR
3025 015726 017746 007652  MOV     @STATUS,-(SP) ;SAVE PRIORITY
3026 015732 012777 016120 007642  MOV     @SEKDON,@VECTOR ;SETUP TRAP FOR RANDOM SEEK
3027 015740 012777 000340 007636  MOV     #340,@STATUS
3028 015746 016777 007746 007642  3$:      MOV     SEEK1,@RPCA ;SET RANDOM CYLINDER
3029 015754 012777 177777 007644  MOV     #-1,@RPDS ;CLEAR ATTENTIONS
3030 015762 042777 000100 007616  BIC     #86,@RPCS ;CLEAR INTERRUPT DONE
3031 015770 052777 020011 007610  BIS     #20011,@RPCS ;SEEK INTERRUPT ON ATTENTION
3032 015776 000001          WAIT
3033 016000 005777 007602          TST     @RPCS ;ERROR?
3034 016004 100035          BPL     4$      ;BR IF NO ERROR
3035 016006 104400          HLT
3036 016010 032777 004000 007570  BIT     #811,@RPCS ;DISK ERROR DURING RANDOM SEEK
3037 016016 001006          BNE     5$      ;SEEK INCOMPLETE?
3038 016020 032777 001000 007500  BIT     #89,@RPDS ;BR IF SEEK INCOMPLETE
3039 016026 001402          BEQ     5$      ;FILE UNSAFE?
3040 016030 000000          HALT ;BR IF NOT UNSAFE
3041 016032 000410          BR      6$      ;FILE UNSAFE
3042 016034          BR      6$      ;GO DO A HOME SEEK
3043 016034 004567 163224  5$:      JSR     RS,PRINT$ ;PRINT MESSAGE
3044 016040 024115          MES14 ;CYLINDER =
3045 016042 016767 007652 163512  MOV     SEEK1,TTY
3046 016050 004767 163302          JSR     PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
3047 016054 012777 000015 007524  6$:      MOV     #15,@RPCS ;DO A HOME SEEK
3048 016062 105777 007520          TSTB   @RPCS
3049 016066 100375          BPL     -4 ;WAIT FOR DONE
3050 016070 005777 007532          TST     @RPDS
3051 016074 100375          BPL     -4 ;WAIT FOR READY
3052 016076 000723          BR      3$      ;TRY SEEK AGAIN
3053 016100 005777 007522  4$:      TST     @RPDS
3054 016104 100375          BPL     -4 ;WAIT FOR READY
3055 016106 012677 007472  MOV     (SP)+,@STATUS ;RESTORE TRAPS
3056 016112 012677 007464  MOV     (SP)+,@VECTOR
3057 016116 000207          SEEND:  RTS     PC ;RETURN FROM RANDOM SEEK
3058 016120 000002          SEKDON: RTI    ;RANDOM SEEK DONE
3059
3060
3061          ;ROUTINE TO SET UP CYLINDER AND DISK ADDRESS FROM
3062          ;OPERATOR INPUTS DURING CONVERSATION MODE.
3063
3064 016122 032767 000040 007502  OPDSEL: BIT     #85,FLAG ;USE OPERATOR ADDR?
3065 016130 001001          BNE     +4
3066 016132 000207          RTS     PC ;NO
3067 016134 016767 007474 007502  MOV     SCYL,CYLINDER ;GET CYLINDER ADDR
3068 016142 016767 007472 007476  MOV     SSEC,DMA ;GET SECTOR ADDR
3069 016150 116767 007462 007471  MOV     SHED,DMA+1 ;GET HEAD ADDR
3070 016156 000207          RTS     PC
3071
3072
3073
3074          ;ROUTINE TO SETUP DISK BUFFERS
3075          ;ADD WORD COUNT TO STARTING DISK ADDRESSES
3076          ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
3077

```

```

3078 016160 032767 000040 007444 DISBUF: BIT #85,FLAG ;DID OPERATOR SUPPLY ADDR?
3079 016166 001401 BEQ .+4
3080 016170 000461 BR BUFEXIT ;OPERATOR DEFINED DISK ADDR
3091 016172 004767 000520 JSR PC,BLSZ ;DEFINE BLOCK SIZE
3092 016176 016767 007462 007500 MOV BLOCK,WORK1
3083 016204 016767 007436 007470 INCSEC: MOV DMA,WORK ;GET DISK ADDR
3084 016212 042767 177760 007462 BIC #177760,WORK ;MASK OUT SECTOR COUNT
3085 016220 022767 000011 007454 CMP #11,WORK ;CHECK FOR LAST SECTOR
3086 016226 001406 BEQ INCSUR ;CHECK SURFACE
3087 016230 005267 007412 INC DMA ;+1 SECTOR COUNT
3088 016234 005367 007424 DECBLK: DEC BLOCK ;-1 FROM BLOCK COUNT
3089 016240 001432 BEQ CMDAE ;CMP DMA TO RPDA
3090 016242 000760 BR INCSEC ;RECYCLE
3091 016244 042767 000017 007374 INCSUR: BIC #17,DMA ;FETCH ADDRESS
3092 016252 016767 007370 007422 MOV DMA,WORK
3093 016260 042767 160377 007414 BIC #160377,WORK
3094 016266 122767 000023 007407 CMPB #23,WORK+1
3095 016274 001403 BEQ SWSUR ;+1 SURFACE
3096 016276 105267 007345 INCB DMA+1 ;INC HEAD NUMBER
3097 016302 000754 BR DECBLK ;RECYCLE
3098 016304 005067 007336 SWSUR: CLR DMA ;CLEAR THE DISK ADDRESS
3099 016310 005267 007330 INC CYLINDER
3100 016314 022767 000626 007322 CMP #626,CYLINDER ;HAS LAST CYL BEEN EXCEEDED?
3101 016322 001404 BEQ BUFEXIT ;BRANCH IF YES
3102 016324 000743 BR DECBLK
3103 ;COME HERE AFTER DETERMINING THE STARTING ADDR OF THE NEXT
3104 ;TRANSFER. NOW CHECK TO SEE THERE IS ENOUGH ROOM ON THE DISK
3105 ;TO MAKE THE TRANSFER. IF NOT MODIFY THE WORD COUNT FOR THE FINAL
3106 ;OUTPUT.
3107
3108
3109 016326 105767 007300 CMDAE: TSTB FLAG ;CHECK FOR LAST DISK BUFFER
3110 016332 100015 BPL BUFINX
3111 016334 005067 007306 BUFEXIT: CLR DMA ;CLEAR ADDRESS BITS
3112 016340 005067 007300 CLR CYLINDER ;CLR CYLINDER REGISTER
3113 016344 062716 000002 ADD #2,(6) ;INC STACK POINTER
3114 016350 042767 000200 007254 BIC #200,FLAG
3115 016356 016767 007274 007256 MOV SWRDCT,WRDCT
3116 016364 000500 BR EXTDR ;EXIT
3117 016366 005067 007314 BUFINX: CLR WORK2 ;CLEAR BLOCK COUNTER
3118 016372 016767 007250 007302 MOV DMA,WORK
3119 016400 016767 007240 007302 MOV CYLINDER,WORK3
3120 016406 042767 160360 007266 BIC #160360,WORK
3121 016414 005267 007266 XINCSEC: INC WORK2 ;INCREMENT BLOCK COUNT
3122 016420 005367 007260 DEC WORK1 ;DECREMENT TOTAL BLOCKS REQUIRED
3123 016424 001460 BEQ EXTDR ;EXIT IF BLOCK COUNT SATISFIED
3124 016426 122767 000011 007246 CMPB #11,WORK ;CHECK THE DISK ADDRESS TO
3125 016434 001403 BEQ XINCSUR ;SEE IF THERE IS ENOUGH ROOM
3126 016436 005267 007240 INC WORK ;TO HANDLE THE OUTPUT REQUESTED
3127 016442 000764 BR XINCSEC
3128 016444 105067 007232 XINCSUR: CLRB WORK
3129 016450 122767 000023 007225 CMPB #23,WORK+1
3130 016456 001403 BEQ 1$
3131 016460 105267 007217 INCB WORK+1
3132 016464 000753 BR XINCSEC
3133 016466 005067 007210 1$: CLR WORK
  
```

```

3134 016472 022767 000625 007210      CMP      #625,WORK3      ;ARE WE ON THE LAST CYLINDER?
3135 016500 001403                      BEQ      25             ;BRANCH IF YES
3136 016502 005267 007202                      INC      WORK3
3137 016506 000742                      BR       XINCSEC
3138 016510 016767 007172 007124 25:    MOV      WORK2,WRDCT    ;COME HERE IF THERE IS NOT
3139 016516 000241                      CLC                               ;ENOUGH ROOM TO HANDLE THE
3140 016520 006167 007116                      ROL      WRDCT          ;REQUESTED OUTPUT. MODIFY THE
3141 016524 006167 007112                      ROL      WRDCT          ;WORDCOUNT TO FILL THE REMAINING
3142 016530 006167 007106                      ROL      WRDCT          ;SURFACE.
3143 016534 006167 007102                      ROL      WRDCT
3144 016540 006167 007076                      ROL      WRDCT
3145 016544 006167 007072                      ROL      WRDCT
3146 016550 006167 007066                      ROL      WRDCT
3147 016554 006167 007062                      ROL      WRDCT
3148 016560 052767 000200 007044      BIS      #200,FLAG
3149 016566 000207                      EXTDR:  RTS      PC      ;EXIT
3150
3151
3152                      ;ROUTINE TO SELECT THE DISK UNIT
3153
3154 016570 016767 007036 007104  DSKNOS: MOV      FLAG,WORK      ;FETCH THE FLAG WORD
3155 016576 006067 007100                      ROR      WORK
3156 016602 006067 007074                      ROR      WORK
3157 016606 000241                      CLC
3158 016610 000367 007066                      SWAB     WORK
3159 016614 042767 174377 007060      BIC      #174377,WORK    ;MASK THE DISK NUMBER
3160 016622 016777 007054 006756      MOV      WORK,DRPCS      ;LOAD THE ADDRESS IN THE ADDRESS REG
3161 016630 005777 006772                      TST      DRPDS
3162 016634 100401                      BMI      15             ;IS THE UNIT READY?
3163 016636 104400                      HLT                               ;BRANCH IF READY
3164 016640 000205                      15:    RTS      RS      ;SELECTED UNIT NOT READY
3165
3166
3167                      ;INITIALIZE THE VECTORS
3168
3169 016642 012767 001112 161164  INIT:  MOV      #ERROR,34      ;SETUP TRAP VECTOR
3170 016650 012767 000340 161160      MOV      #PRI7,36
3171 016656 012767 001004 161144      MOV      #SCOPE$,30      ;SETUP EMT VECTOR
3172 016664 012767 000340 161140      MOV      #PRI7,32
3173 016672 012777 015224 006702      MOV      #DKINT,VECTOR    ;SETUP DISK INTERRUPT VECTOR
3174 016700 012777 000340 006676      MOV      #PRI7,STATUS
3175 016706 012737 000340 177776      MOV      #PRI7,#PSW      ;LOCKOUT INTERRUPTS
3176 016714 000207                      RTS      PC
3177
3178
3179                      ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
3180
3181 016716 012767 000377 006740  BLSZ:  MOV      #377,BLOCK    ;DRIVE BLOCK SIZE
3182 016724 016767 006712 006750      MOV      WRDCT,WORK      ;FETCH WORD COUNT
3183 016732 036767 006726 006742      BIT      BLOCK,WORK
3184 016740 001410                      BEQ      RORBLK
3185 016742 046767 006716 006732      BIC      BLOCK,WORK      ;SET UP BLOCK OVERFLOW
3186 016750 005267 006710                      INC      BLOCK
3187 016754 066767 006704 006720      ADD      BLOCK,WORK
3188 016762 000367 006714                      RORBLK: SWAB     WORK
3189 016766 016767 006710 006670      MOV      WORK,BLOCK      ;BLOCK COUNT

```

```

3190 016774 000207          RIS      PC          ;EXIT
3191
3192
3193          ; DETERMINE THE APPROPRIATE ATTENTION BIT FROM
3194          ; THE UNIT NUMBER.
3195
3196 016776 016701 006630    GATTN:  MOV     FLAG,R1
3197 017002 006001          ROR     R1
3198 017004 006001          ROR     R1          ;GET UNIT NUMBER
3199 017006 005067 000014    CLR     ATTN
3200 017012 042701 177770    BIC     #177770,R1 ;ISOLATE UNIT
3201 017016 116167 017030 000002  MOVB   ATTNB(R1),ATTN ;GET ATTENTION BIT
3202 017024 000207          RTS     PC
3203
3204
3205 017026 000000          ATTN:   0
3206 017030      001      002      004  ATTNB:  .BYTE  1,2,4,10,20,40,100,200
3207 017033      010      020      040
3208 017036      100      200
3209          .EVEN
3210
3211
3212
3213
3214          ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
3215
3216          ;ENTER FROM JSR PC PASEL
3217 017040 016700 006606    PASEL:  MOV     PATNJ,RO ;SET UP PATTERN NUMBER
3218 017044 016767 006572 006630  MOV     WRDCT,WORK ;SET UP WORK
3219 017052 012701 025730    MOV     #OUTBUF,R1 ;LOC. OF OUT BUFFER
3220 017056 032767 000500 006546  BIT     #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM.MANAG.
3221 017064 001002          BNE     IS ;BR IF TRUE
3222 017066 012701 040000    MOV     #40000,R1 ;START OF MEM. MANAG. BUFFER
3223 017072 022700 000034    IS:    CMP     #34,RO ;TEST FOR RANDOM DATA NUMBER
3224 017076 001423          BEQ     RANDOM ;GO GENERATE RANDOM DATA
3225 017100 022700 000032    CMP     #32,RO ;IS THIS PATTERN IS
3226 017104 001406          BEQ     PATT32
3227 017106 016021 017762    FILDAT: MOV     PATD(0),(1)+ ;FILL BUFFER
3228 017112 005367 006564    DEC     WORK ;DEC. WORK COUNT
3229 017116 001373          BNE     FILDAT ;LOAD NEXT WORD
3230 017120 000207          RTS     PC ;BUFFER FULL
3231 017122 012721 177777    PATT32: MOV     #177777,(1)+ ;INSERT ALL ONES PATTERN
3232 017126 005367 006550    DEC     WORK
3233 017132 001404          BEQ     IS
3234 017134 005021          CLR     (1)+ ;LOAD ZERO PATTERN
3235 017136 005367 006540    DEC     WORK ;DECREMENT WORD COUNT
3236 017142 001367          BNE     PATT32 ;LOOP IF NOT ZERO
3237 017144 000207          RTS     PC ;EXIT
3238
3239 017146 016767 000134 000136  ;RANDOM DATA GENERATOR SUBROUTINE
3240 017154 016767 000130 000132  RANDOM: MOV     LONUN,LOSAY
3241 017162 016700 000120          MOV     HINUN,HISAV
3242 017166 016704 000116    IS:    MOV     LONUN,RO ;SET UP RO WITH 5 DIGITS LOW
3243 017172 012703 000007    MOV     HINUN,R4 ;SET UP R1 WITH 5 DIGITS HIGH
3244 017176 005002          MOV     #7,R3 ;SET UP SHIFT COUNT
3245 017200 006300          CLR     R2 ;CLEAR R2
          ASL     RO ;SHIFT RO LEFT AND
  
```

3246	017202	006104			ROL	R4	; ROTATE CARRY INTO LSB OF R1 INTO	
3247	017204	006102			ROL	R2	; ROTATE CARRY OUT OF R1 INTO R2	
3248	017206	005303			DEC	R3	; DECREMENT R3	
3249	017210	001373			BNE	2\$	; CONTINUE SHIFT LOOP	
3250	017212	066702	000070		ADD	LONUN, R2	; ADDN IN NUMBER TO MAKE X 129	
3251	017216	005504			ADC	R4	; PROPOGATE CARRY	
3252	017220	066704	000064		ADD	HINUN, R4	; ADDN IN NUMBER TO MAKE X 129	
3253	017224	005502			ADC	R2	; PROPOGATE CARRY	
3254	017226	062700	001057		ADD	#1057, R0	; ADDN LOW CONSTANT	
3255	017232	005504			ADC	R4	; PROPOGATE CARRIES	
3256	017234	005502			ADC	R2	; PROPOGATE AGAIN	
3257	017236	062704	047401		ADD	#47401, R4	; ADDN HIGH CONSTANT	
3258	017242	005502			ADC	R2	; PROPOGATE CARRY	
3259	017244	062702	000006		ADD	#6, R2	; ADDN HIGHEST CONSTANT	
3260	017250	060200			ADD	R2, R0	; REPRIME R0 WITH HIGH DIGIT	
3261	017252	005504			ADC	R4	; PROPOGATE CARRY	
3262	017254	010067	000026		MOV	R0, LONUN	; PUT R0 BACK IN LONUM	
3263	017260	010021			MOV	R0, (1)+	; HOLD LONUM FOR PROGRAM	
3264	017262	005367	006414		DEC	WORK		
3265	017266	001406			BEQ	EXGEN		
3266	017270	010467	000014		MOV	R4, HINUN	; PUT R1 BACK IN HINUM	
3267	017274	010421			MOV	R4, (1)+	; HOLD HINUM FOR PROGRAM	
3268	017276	005367	006400		DEC	WORK		
3269	017302	001327			BNE	1\$		
3270	017304	000207		EXGEN:	RTS	PC	; RETURN TO PROGRAM	
3271	017306	000000		LONUN:	0			
3272	017310	000000		HINUN:	0			
3273	017312	000000		LOSAY:	0			
3274	017314	000000		HISAY:	0			
3275								
3276								
3277	017316	032767	000002	161716	MSG:	BIT	#B1, HLTCTS	; TYPE ENTIRE MESSAGE
3278	017324	001103			BNE	1\$		; BRANCH IF NO
3279	017326	004567	161732		JSR	R5, PRINTS		; PRINT MESSAGE
3280	017332	024002			MESB			
3281	017334	004567	161724		JSR	R5, PRINTS		; PRINT MESSAGE
3282	017340	023611			MES2A			
3283	017342	017767	006260	162212	MOV	ARPOS, TTY		
3284	017350	004767	161770		JSR	PC, PRINTR		; TYPE LOCATION WITH LEADING ZEROS
3285	017354	004567	161704		JSR	R5, PRINTS		; PRINT MESSAGE
3286	017360	023567			MES1A			
3287	017362	017767	006236	162172	MOV	ARPER, TTY		
3288	017370	004767	161750		JSR	PC, PRINTR		; TYPE LOCATION WITH LEADING ZEROS
3289	017374	004567	161664		JSR	R5, PRINTS		; PRINT MESSAGE
3290	017400	023600			MES2			
3291	017402	017767	006200	162152	MOV	ARPCS, TTY		
3292	017410	004767	161730		JSR	PC, PRINTR		; TYPE LOCATION WITH LEADING ZEROS
3293	017414	004567	161644		JSR	R5, PRINTS		; PRINT MESSAGE
3294	017420	023622			MES2B			
3295	017422	017767	006170	162132	MOV	ARPCA, TTY		
3296	017430	004767	161710		JSR	PC, PRINTR		; TYPE LOCATION WITH LEADING ZEROS
3297	017434	004567	161624		JSR	R5, PRINTS		; PRINT MESSAGE
3298	017440	023633			MES2C			
3299	017442	017767	006152	162112	MOV	ARPOA, TTY		
3300	017450	004767	161670		JSR	PC, PRINTR		; TYPE LOCATION WITH LEADING ZEROS
3301	017454	004567	161604		JSR	R5, PRINTS		; PRINT MESSAGE

3302	017460	023644			MES20		
3303	017462	017767	006142	162072	MOV	@SUCA, TTY	
3304	017470	004767	161650		JSR	PC, PRINTR	; TYPE LOCATION WITH LEADING ZEROS
3305	017474	022767	000005	006174	CMP	#5, TESTNO	; TEST 5?
3306	017502	001404			BEQ	4\$	; BR IF YES
3307	017504	022767	000006	006164	CMP	#6, TESTNO	; TEST 6?
3308	017512	001010			BNE	1\$	; BR IF NO
3309	017514				4\$:		
3310	017514	004567	161544		JSR	R5, PRINT\$	; PRINT MESSAGE
3311	017520	024230			MES21		; READ COUNTER =
3312	017522	016767	006174	162032	MOV	CNTA, TTY	
3313	017530	004767	161622		JSR	PC, PRINTS	; TYPE LOCATION-SUPRESS ZEROS
3314	017534	032767	000001	161500	1\$:	BIT	#B0, HLTCTS
3315	017542	001001			BNE	2\$	; TYPE EXP-REC
3316	017544	000207			RTS	PC	; BRANCH IF YES
3317	017546	032767	000002	161466	2\$:	BIT	#B1, HLTCTS
3318	017554	001403			BEQ	3\$	
3319	017556	004567	161502		JSR	R5, PRINT\$	; PRINT MESSAGE
3320	017562	024147			MES17		
3321	017564	032767	000004	161450	3\$:	BIT	#B2, HLTCTS
3322	017572	001450			BEQ	5\$	; TYPE MEMORY MANAGEMENT REGISTERS?
3323	017574	004567	161464		JSR	R5, PRINT\$	; BR IF NO
3324	017600	024327			MES23		; PRINT MESSAGE
3325	017602	013767	172344	161752	MOV	@#KIPAR2, TTY	
3326	017610	004767	161542		JSR	PC, PRINT\$	; TYPE LOCATION-SUPRESS ZEROS
3327	017614	004567	161444		JSR	R5, PRINT\$	; PRINT MESSAGE
3328	017620	024344			MES24		
3329	017622	013767	172346	161732	MOV	@#KIPAR3, TTY	
3330	017630	004767	161522		JSR	PC, PRINT\$	; TYPE LOCATION-SUPRESS ZEROS
3331	017634	004567	161424		JSR	R5, PRINT\$	; PRINT MESSAGE
3332	017640	024367			MES25		
3333	017642	013767	172350	161712	MOV	@#KIPAR4, TTY	
3334	017650	004767	161502		JSR	PC, PRINT\$	; TYPE LOCATION-SUPRESS ZEROS
3335	017654	004567	161404		JSR	R5, PRINT\$	; PRINT MESSAGE
3336	017660	024404			MES26		
3337	017662	013767	172352	161672	MOV	@#KIPAR5, TTY	
3338	017670	004767	161462		JSR	PC, PRINT\$	; TYPE LOCATION-SUPRESS ZEROS
3339	017674	004567	161364		JSR	R5, PRINT\$	; PRINT MESSAGE
3340	017700	024427			MES27		
3341	017702	013767	172354	161652	MOV	@#KIPAR6, TTY	
3342	017710	004767	161442		JSR	PC, PRINT\$	; TYPE LOCATION-SUPRESS ZEROS
3343	017714				5\$:		
3344	017714	004567	161344		JSR	R5, PRINT\$	; PRINT MESSAGE
3345	017720	024162			MES18		
3346	017722	016767	000030	161632	MOV	EXPS, TTY	
3347	017730	004767	161410		JSR	PC, PRINTR	; TYPE LOCATION WITH LEADING ZEROS
3348	017734	004567	161324		JSR	R5, PRINT\$	; PRINT MESSAGE
3349	017740	024176			MES19		
3350	017742	016767	000012	161612	MOV	RECS, TTY	
3351	017750	004767	161370		JSR	PC, PRINTR	; TYPE LOCATION WITH LEADING ZEROS
3352	017754	000207			RTS	PC	
3353	017756	000000			EXPS:	0	
3354	017760	000000			RECS:	0	
3355							
3356							
3357							

```

3358                                     .EVEN
3359                                     ;RP11 DATA PATTERNS
3360
3361 017762 163126 PAT0: 163126
3362 017764 052525 PAT1: 052525
3363 017766 125252 PAT2: 125252
3364 017770 031463 PAT3: 031463
3365 017772 007417 PAT4: 007417
3366 017774 010421 PAT5: 010421
3367 017776 021042 PAT6: 021042
3368 020000 042104 PAT7: 042104
3369 020002 104210 PAT10: 104210
3370 020004 167356 PAT11: 167356
3371 020006 156735 PAT12: 156735
3372 020010 135673 PAT13: 135673
3373 020012 073567 PAT14: 073567
3374 020014 000001 PAT15: 000001
3375                                     ;PAT16 RANDOM DATA
3376
3377                                     ;THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
3378                                     ;ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 5 OF THE SWITCH
3379                                     ;REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN
3380                                     ;ERROR HAS BEEN FOUND AND WILL REPORT UP TO 3 VERIFY ERRORS
3381                                     ;WITHIN THE SAME INPUT OPERATION.
3382 020016 012767 177775 005654 COMPAR: MOV #-3,ERCOUNT ;ERROR RETRY COUNTER
3383 020024 016767 005612 005654 MOV WRDCT,WORK2 ;GET THE WORD COUNT
3384 020032 012767 025730 005620 MOV #OUTBUF,SAVE ;SET UP OUTBUFFER POINTER
3385 020040 032767 000500 005564 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG
3386 020046 001003 BNE IS ;BR IF TRUE
3387 020050 012767 040000 005602 MOV #40000,SAVE ;START OF MEM. MANAG. OUT BUFFER
3388 020056 005067 005566 IS: CLR SWITCH ;CLEAR RANDOM PATTERN FLAG
3389 020062 016767 177224 161612 MOV LOSAV,LONUM ;GET RANDOM BASE NOS.
3390 020070 016767 177220 161602 MOV HISAV,HINUM
3391 020076 022767 000034 005546 CMP #34,PATNU ;IS THIS RANDOM PATTERN?
3392 020104 001422 BEQ CMPLP ;BRANCH IF YES
3393 020106 022767 000032 005536 CMP #32,PATNU ;IS THIS SPECIAL PATTERN?
3394 020114 001037 BNE CMPLP1 ;BRANCH IF NO
3395 020116 005767 005526 CMPLP2: TST SWITCH
3396 020122 001003 BNE IS
3397 020124 012767 177777 177624 MOV #177777,EXPS ;EXPECT ALL ONES
3398 020132 010667 005512 MOV SP,SWITCH ;SET THE FLAG
3399 020136 000433 BR WRDCMP ;GO COMPARE DATA
3400 020140 005067 005504 IS: CLR SWITCH
3401 020144 005067 177606 CLR EXPS ;EXPECT ALL ZEROS
3402 020150 000426 BR WRDCMP ;GO COMPARE DATA
3403 020152 005767 005472 CMPLP: TST SWITCH
3404 020156 001010 BNE 2S
3405 020160 004767 161400 JSR PC,RANDS ;GENERATE TWO RANDOM NOS.
3406 020164 016767 161512 177564 MOV LONUM,EXPS ;GET EVEN RANDOM WORD
3407 020172 010667 005452 MOV SP,SWITCH ;SET RANDOM PATTERN FLAG
3408 020176 000413 BR WRDCMP
3409 020200 005067 005444 2S: CLR SWITCH
3410 020204 016767 161470 177544 MOV HINUM,EXPS
3411 020212 000405 BR WRDCMP
3412 020214 016700 005432 CMPLP1: MOV PATNU,RO
3413 020220 016067 017762 177530 MOV PAT0(RO),EXPS

```

3414	020226	027767	005426	177522	WRDCMP:	CMP	SAVE,EXPS	;COMPARE DATA
3415	020234	001021				BNE	WDERR	;WORD IN ERROR
3416	020236	005367	005444		WRDINC:	DEC	WORK2	;DECREMENT THE WORD COUNT
3417	020242	001415				BEQ	ADAM	;EXIT ROUTINE IF ZERO
3418	020244	062767	000002	005406	BLAD1:	ADD	#2,SAVE	;UPDATE PATTERN ADDRESS
3419	020252	022767	000032	005372		CMP	#32,PATNU	
3420	020260	101362				BHI	WRDCMP	;BRANCH IF STANDARD PATTERN
3421	020262	022767	000034	005362		CMP	#34,PATNU	;IS THIS RANDOM PATTERN
3422	020270	001730				BEQ	CMPLP	;BRANCH IF YES
3423	020272	000711				BR	CMPLP2	;BRANCH IF YES
3424	020274	000754				BR	WRDCMP	;COMPARE NEXT WORD
3425	020276	000207			ADAM:	RTS	PC	;EXIT THIS ROUTINE
3426	020300	005767	164174		WDERR:	TST	INTFLG	;DID INTERRUPT OCCUR YET?
3427	020304	001750				BEQ	WRDCMP	;BRANCH IF NO
3428	020306	017767	005346	177444		MOV	SAVE,RECS	;GET GOOD DATA
3429	020314	010667	160720			MOV	SP,ERRFLG	;SET ERROR FLAG
3430	020320	005767	005346			TST	RDERR	;IS THIS THE FIRST READ ERROR?
3431	020324	001404				BEQ	3\$	;BRANCH IF YES
3432	020326	032737	000020	177570		BIT	#B4,#SWR	;PRINT ALL RETRY ERRORS?
3433	020334	001556				BEQ	1\$	;BRANCH IF NO
3434	020336	032767	000100	005266	3\$:	BIT	#B6,FLAG	;MEMORY MANAGEMENT?
3435	020344	001002				BNE	10\$	;BR IF NO
3436	020346	104405				HLT	+5	;DATA COMPARE ERROR USING MEM. MANAG.
3437	020350	000401				BR	9\$	;GET AROUND HLT+3
3438	020352	104403			10\$:	HLT	+3	;DATA COMPARE ERROR
3439	020354	005067	005304		9\$:	CLR	BLOCK	;CLEAR THE BLOCK COUNTER
3440	020360	016767	005256	005314		MOV	WRDCT,WORK	;GET THE WORD COUNT
3441	020366	166767	005314	005306		SUB	WORK2,WORK	;DETERMINE DISTANCE OF FAILURE INTO BUFFER
3442	020374	162767	000400	005300	2\$:	SUB	#400,WORK	
3443	020402	100403				BMI	8\$	
3444	020404	005267	005254			INC	BLOCK	;UPDATE BLOCK COUNT FOR EACH 400 WORDS
3445	020410	000771				BR	2\$	
3446	020412	062767	000400	005262	8\$:	ADD	#400,WORK	;RESTORE POSITIVE NUMBER
3447	020420	016767	005222	005256		MOV	DMA,WORK1	;GET HEAD AND SECTOR ADDRESS
3448	020426	016767	005212	005254		MOV	CYLINDER,WORK3	;GET CYLINDER ADDRESS
3449	020434	005767	005224		5\$:	TST	BLOCK	;IS THE BLOCK COUNT ZERO?
3450	020440	001427				BEQ	7\$	;BRANCH IF YES
3451	020442	005367	005216			DEC	BLOCK	;DECREMENT BLOCK COUNT
3452	020446	122767	000011	005230		CMPB	#11,WORK1	;DETERMINE THE CYLINDER, HEAD,
3453	020454	001403				BEQ	4\$	;AND SECTOR ADDRESSES OF THE
3454	020456	005267	005222			INC	WORK1	;COMPARE ERROR
3455	020462	000764				BR	5\$	
3456	020464	105067	005214		4\$:	CLRB	WORK1	
3457	020470	122767	000023	005207		CMPB	#23,WORK1+1	
3458	020476	001403				BEQ	6\$	
3459	020500	105267	005201			INCB	WORK1+1	
3460	020504	000753				BR	5\$	
3461	020506	005067	005172		6\$:	CLR	WORK1	
3462	020512	005267	005172			INC	WORK3	
3463	020516	000746				BR	5\$	
3464	020520				7\$:			
3465	020520	004567	160540			JSR	R5,PRINT\$	;PRINT MESSAGE
3466	020524	024115				MES14		;GIVE CYL ADDR
3467	020526	016767	005156	161026		MOV	WORK3,TTY	
3468	020534	004767	160616			JSR	PC,PRINT\$	;TYPE LOCATION-SUPPRESS ZEROS
3469	020540	005067	005130			CLR	ACNVX	



```

3470 020544 116767 005135 005122      MOVB   WORK1+1,ACNVX
3471 020552 004567 160506      JSR    RS,PRINT$      ;PRINT MESSAGE
3472 020556 024125      MES15                    ;GIVE HEAD ADDR
3473 020560 016767 005110 160774      MOV    ACNVX,TTY
3474 020566 004767 160564      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
3475 020572 116767 005106 005074      MOVB   WORK1,ACNVX
3476 020600 004567 160460      JSR    RS,PRINT$      ;PRINT MESSAGE
3477 020604 024136      MES16                    ;GIVE SECTOR ADDR
3478 020606 016767 005062 160746      MOV    ACNVX,TTY
3479 020614 004767 160536      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
3480 020620 004567 160440      JSR    RS,PRINT$      ;PRINT MESSAGE
3481 020624 024015      MES9
3482 020626 016767 005050 005040      MOV    WORK,ACNVX      ;GET WORD COUNT INTO SECTOR
3483 020634 005267 005034      INC    ACNVX
3484 020640 016767 005030 160714      MOV    ACNVX,TTY
3485 020646 004767 160504      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
3486 020652 004567 160406      JSR    RS,PRINT$      ;PRINT MESSAGE
3487 020656 024101      MES13
3488 020660 016767 005006 160674      MOV    RDERR,TTY
3489 020666 004767 160464      JSR    PC,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
3490 020672 032737 000040 177570 1$:  BIT    #B5,#$SWR      ;CONTINUE COMPARING?
3491 020700 001405      BEQ    11$             ;BRANCH IF NO
3492 020702 005267 004772      INC    ERCOUNT      ;UPDATE ERROR COUNTER
3493 020706 001402      BEQ    11$
3494 020710 000167 177322      JMP    WRDINC
3495 020714 000167 177356      11$:  JMP    ADAM
3496
3497
3498
3499
3500      ;EXTENDED MEMORY EXERCISER
3501      ;THE PROGRAM DETERMINES HOW MUCH MEMORY
3502      ;IS ON THE SYSTEM THEN IT
3503      ;GENERATES A RANDOM BUFFER THAT SIZE
3504      ;AND WRITES AND WRITE CHECKS THE DATA
3505 020720 032767 000100 004704  EXTMEN: BIT    #B6,FLAG      ;MEMORY MANAGEMENT IN USE?
3506 020726 001002      BNE    1$             ;BR IF NO
3507 020730 005037 177572      CLR    #SRO           ;TURN IT OFF
3508 020734 052777 000001 004644  1$:  BIS    #B0,#RPCS      ;CLEAR THE DISK
3509 020742 105777 004640      TSTB   #RPCS
3510 020746 100375      BPL    -4
3511 020750 012737 000340 177776      MOV    #PRI7,#PSW     ;LOCK UP PRIORITY LEVELS
3512 020756 012767 021026 157020      MOV    #MAXREF,4      ;SET UP I/O BUS TRAP
3513 020764 012767 000340 157014      MOV    #PRI7,6
3514 020772 012767 017446 004660      MOV    #17446,SAVE    ;SET UP FOR 4K
3515 021000 005777 004654      EXREF: TST   #SAVE     ;REFERENCE MEMORY
3516 021004 022767 157446 004646      CMP    #157446,SAVE   ;TEST FOR 28K
3517 021012 001001      BNE    1$             ;BRANCH IF LESS THAN 28K
3518 021014 000407      BR     MAXRF1         ;LAST REFERENCE MADE TO I/O REG.
3519 021016 062767 020000 004634  1$:  ADD    #20000,SAVE    ;SET UP FOR NEXT MEMORY REF.
3520 021024 000765      BR     EXREF          ;GO REFERENCE MEMORY
3521
3522      ;ENTER HERE WHEN I/O BUS ERROR OCCURS
3523
3524 021026 162767 020000 004624  MAXREF: SUB   #20000,SAVE
3525 021034 012767 000006 156742  MAXRF1: MOV   #6,4      ;RESTORE I/O BUS TRAP

```

```

3526 021042 005067 156740 CLR 6
3527 021046 005737 000042 TST @#42 ; UNDER MONITOR CONTROL?
3528 021052 001403 BEQ 1$ ; BRANCH IF NO
3529 021054 162767 005670 004576 SUB #3000,SAVE ; ALLOW ROOM FOR THE MONITOR
3530 021062 016767 004572 000660 1$: MOV SAVE,MMSIZ ; SAVE THE MAXIMUM MEMORY ADDRESS
3531 021070 004767 000070 JSR PC,MMSIZE ; GO SIZE WITH MEMORY MANAGEMENT
3532 021074 162767 025730 004556 MMBK: SUB #OUTBUF,SAVE ; DETERMINE THE BUFFER SIZE
3533 021102 000241 CLC
3534 021104 006067 004550 ROR SAVE ; FORM WORD COUNT
3535 021110 016767 004544 004524 MOV SAVE,WRDCT ; SAVE IT
3536 021116 042767 000001 004534 BIC #80,SAVE ; MAKE ADDRESS EVEN
3537 021124 012767 025730 004536 MOV #OUTBUF,INBUF ; START OF INPUT BUFFER
3538 021132 066767 004522 004530 ADD SAVE,INBUF
3539 021140 000241 CLC
3540 021142 042767 000377 004472 BIC #377,WRDCT ; DETERMINE MAXIMUM WORD COUNT
3541 021150 016767 004466 004500 MOV WRDCT,SWRDCT
3542 021156 012706 000476 EXIT: MOV #STKPTR-2,SP
3543 021162 000205 RTS RS
3544
3545 ; THIS SUB-ROUTINE MAPS FOR MEMORY MANAGEMENT AND SETS UP
3546 ; BUFFER SIZES WITH MEMORY MANAGEMENT
3547
3548 021164 032767 000100 004440 MMSIZE: BIT #86,FLAG ; USING MEMORY MANAGEMENT?
3549 021172 001340 BNE MMBK ; BR IF NO
3550 021174 005237 177572 INC @#SRO ; TURN MEMORY MANAGEMENT BACK ON
3551 021200 012737 021246 000004 MOV #NXM,@#ERRVEC ; SET UP FOR TRAP
3552 021206 012737 000340 000006 MOV #340,@#ERRVEC+2
3553 021214 012737 000400 172344 MOV #400,@#KIPAR2 ; SETUP FOR NEXT 4K PAGE
3554 021222 012737 177406 172304 MOV #400*256.-400+UP,RW,@#KIPDR2 ; SET KIPDR2=RW UP 400 BLOCKS
3555 021230 012700 040000 MOV #40000,RO ; SET UP FOR TEST
3556 021234 005710 1$: TST (RO)
3557 021236 062737 000040 172344 ADD #40,@#KIPAR2 ; JUMP PAGE BY 1K
3558 021244 000773 BR 1$ ; TEST NEXT PAGE
3559 021246 022737 000400 172344 NXM: CMP #400,@#KIPAR2 ; BK MACHINE?
3560 021254 001013 BNE 1$ ; BR IF MORE THAN BK
3561 021256 052767 000400 004345 BIS #88,FLAG ; BK MACHINE WITH MEMORY MANAGEMENT
3562 021264 005037 172304 CLR @#KIPDR2 ; SET KIPDR2 TO BE NON EXISTANT
3563 021270 012737 000006 000004 MOV #6,@#4 ; RESTORE TRAP CATCHER
3564 021276 005037 000006 CLR @#6
3565 021302 000674 BR MMBK ; FINISH MAPPING AS IF NO MEMORY MANAGEMENT
3566 021304 013767 172344 000446 1$: MOV @#KIPAR2,PARMAX ; PUT KIPAR2 INTO WORK LOCATION
3567 021312 032767 000340 000440 BIT #340,PARMAX ; EVEN 4K?
3568 021320 001435 BEQ 4$ ; BR IF EVEN 4K BLOCK
3569 021322 032767 000140 000430 BIT #140,PARMAX ; LAST PAGE 3K?
3570 021330 001422 BEQ 3$ ; BR IF YES
3571 021332 032767 000100 000420 BIT #100,PARMAX ; LAST PAGE 2K?
3572 021340 001407 BEQ 2$ ; BR IF YES
3573 021342 162767 000040 000410 SUB #40,PARMAX ; CREAT MAX NUMBER FOR KIPAR
3574 021350 012767 002000 000376 MOV #2000,MMSIZ ; MAX NUMBER OF WORDS FOR LAST PAGE
3575 021356 000424 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS
3576 021360 162767 000100 000372 2$: SUB #100,PARMAX ; CREAT MAX NUMBER FOR KIPAR
3577 021366 012767 004000 000360 MOV #4000,MMSIZ ; MAX NUMBER OF WORDS IN LAST PAGE
3578 021374 000415 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS
3579 021376 162767 000140 000354 3$: SUB #140,PARMAX ; CREAT MAX NUMBER FOR KIPAR
3580 021404 012767 006000 000346 MOV #6000,PARMAX ; MAX NUMBER OF WORDS IN LAST PAGE
3581 021412 000406 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS

```

3582	021414	162767	000200	000336	4S:	SUB	#200,PARMAX	;CREAT MAX NUMBER FOR KIPAR
3583	021422	012767	010000	000324		MOV	#10000,MMSIZ	;MAX NUMBER OF WORDS IN LAST PAGE
3584	021430	022767	001600	000322	MMFIN:	CMP	#1600,PARMAX	;PARMAX >OR= 24K
3585	021436	003436				BLE	1\$	;BR IF YES
3586	021440	012700	000600			MOV	#600,RO	;SET UP FOR NUMBER OF 4K PAGES
3587	021444	012701	000001			MOV	#1,R1	
3588	021450	020167	000304		2S:	CMP	R1,PARMAX	;DO WE HAVE THE NUMBER OF PAGES
3589	021454	001404				BEQ	3\$	;BR IF YES
3590	021456	062700	000200			ADD	#200,RO	;INCREMENT PAGE TEST
3591	021462	005201				INC	R1	;INCREMENT NUMBER OF PAGES
3592	021464	000771				BR	2\$	;TEST MORE
3593	021466	005067	004150		3S:	CLR	WRDCT	;CLEAR WORD COUNT
3594	021472	010167	000260			MOV	R1,PDRS	;STORE NUMBER OF KIPDRS
3595	021476	062767	010000	004136	4S:	ADD	#10000,WRDCT	;4K BUFFER
3596	021504	005301				DEC	R1	;REDUCE NUMBER OF 4K PAGES
3597	021506	001373				BNE	4\$	;BR IF MORE 4K PAGES
3598	021510	066767	000240	004124		ADD	MMSIZ,WRDCT	;ADD SIZE OF LAST PAGE
3599	021516	016767	004120	004132		MOV	WRDCT,SWRDC	;SAVE TOTAL BUFFER SIZE
3600	021524	016767	004112	000230		MOV	WRDCT,MAXWC	;SAVE AGAIN FOR CONFORMITY
3601	021532	000414				BR	SALDR	;GO SAVE THE LOADERS
3602	021534	012767	050000	000220	1S:	MOV	#50000,MAXWC	;20K TRANSFERS
3603	021542	012767	040000	004106		MOV	#40000,SWRDC	;SET UP HIGHEST TRANSFER SIZE
3604	021550	066767	000200	004100		ADD	MMSIZ,SWRDC	;ADD LAST PAGE SIZE
3605	021556	012767	000005	000172		MOV	#5,PDRS	;SET UP COUNT FOR KIPDRS
3606	021564	005767	004134		SALDR:	TST	LDRFLG	;ARE THE LOADERS ALREADY RELOCATED?
3607	021570	001017				BNE	1\$	;BR IF YES
3608	021572	005037	177572			CLR	#SR0	;TURN OFF MEMORY MANAGEMENT
3609	021576	005167	004122			COM	LDRFLG	;ADJUST LOADER RELOCATED FLAG
3610	021602	016700	000142			MOV	MEMSIZ,RO	
3611	021606	042700	003777			BIC	#3777,RO	;SET UP FOR 1K WORD
3612	021612	012737	021630	000004		MOV	#1\$,#ERRVEC	;SETUP TRAP CATCHER
3613	021620	012701	034000			MOV	#34000,R1	;8K START ADDRESS FOR LOADERS
3614	021624	012021			2S:	MOV	(R0)+,(R1)+	;MOVE LOADER
3615	021626	000776				BR	2\$	
3616	021630	012737	000006	000004	1S:	MOV	#6,#ERRVEC	;RESTORE TRAPCATCHER
3617	021636	005037	000006			CLR	#ERRVEC+2	
3618	021642	052737	000001	177572		BIS	#80,#SR0	;TURN MEMORY MANAGEMENT BACK ON IF OFF
3619	021650	016700	000102			MOV	PDRS,RO	;PUT NUMBER OF KIPDRS INTO WORK REGISTER
3620	021654	005300				DEC	RO	;SETUP FOR PROPER NUMBER OF KIPDRS
3621	021656	001422				BEQ	3\$	;BR IF NO KIPDRS NEED SETTING UP
3622	021660	012737	177406	172306		MOV	#400*256.-400+UP,RW,#KIPDR3	;SET KIPDR3=RW UP 400 BLOCKS
3623	021666	005300				DEC	RO	;ENOUGH KIPDRS?
3624	021670	001415				BEQ	3\$	;BR IF ENOUGH
3625	021672	012737	177406	172310		MOV	#400*256.-400+UP,RW,#KIPDR4	;SET KIPDR4=RW UP 400 BLOCKS
3626	021700	005300				DEC	RO	;ENOUGH KIPDRS?
3627	021702	001410				BEQ	3\$	;BR IF YES
3628	021704	012737	177406	172312		MOV	#400*256.-400+UP,RW,#KIPDR5	;SET KIPDR5=RW UP 400 BLOCKS
3629	021712	005300				DEC	RO	;ENOUGH KIPDRS?
3630	021714	001403				BEQ	3\$	;BR IF YES
3631	021716	012737	177406	172314		MOV	#400*256.-400+UP,RW,#KIPDR6	;SET KIPDR6=RW UP 400 BLOCKS
3632	021724				3S:			
3633	021724	004567	157352			JSR	R5,PRNTPS	;FORCE PRINT THE MESSAGE
3634	021730	024250				MES22		;LOADERS HAVE BEEN RELOCATED TO 8K
3635								;TO RESTORE LOADERS START AT LOCATION
3636	021732	012767	023466	157622		MOV	#RELOAD,TTY	
3637	021740	004767	157434			JSR	PC,PRINTB	;FORCE TYPE LOCATION - SUPPRESS ZEROS

```

3638 021744 000167 177206          JMP      EXIT          ;MEMORY MANAGEMENT DONE
3639
3640 021750 000000          MEMSIZ: 0
3641 021752 000900          MEX:      0          ;CONTROLLER EXTENDED MEMORY ADDRESS
3642 021754 000000          MMSIZ: 0          ;THE LAST PAGE SIZE
3643 021756 000000          PDRS: 0          ;THE NUMBER OF PAGES IN THE BUFFER
3644 021760 000000          PARMAX: 0        ;THE MAXIMUM KIPDR VALUE
3645 021762 000000          MAXWC: 0
3646
3647          ;SUB-ROUTINE TO INITIALIZE KIPAR2-6
3648
3649 021764 016700 177766          PARINT: MOV      PDRS,RO          ;GET NUMBER OF KIPARS
3650 021770 012737 000400 172344          MOV      #400,2#KIPAR2        ;SETUP FIRST PAGE
3651 021776 005300          DEC      RO                  ;DONE?
3652 022000 001422          BEQ      1$                  ;BR IF YES
3653 022002 012737 000600 172346          MOV      #600,2#KIPAR3        ;SET UP SECOND PAGE
3654 022010 005300          DEC      RO                  ;DONE?
3655 022012 001415          BEQ      1$                  ;BR IF YES
3656 022014 012737 001000 172350          MOV      #1000,2#KIPAR4       ;SETUP THIRD PAGE
3657 022022 005300          DEC      RO                  ;DONE?
3658 022024 001410          BEQ      1$                  ;BR IF YES
3659 022026 012737 001200 172352          MOV      #1200,2#KIPAR5       ;SETUP FOURTH PAGE
3660 022034 005300          DEC      RO                  ;DONE?
3661 022036 001403          BEQ      1$                  ;BR IF YES
3662 022040 012737 001400 172354          MOV      #1400,2#KIPAR6       ;SETUP FIFTH PAGE
3663 022046 062716 000004          1$:      ADD      #4,(SP)
3664 022052 000207          RTS      PC
3665
3666          ;SUB-ROUTINE TO INCREMENT EACH KIPAR USED BY 1K
3667
3668 022054 005000          PARINC: CLR      RO          ;SETUP INDEX
3669 022056 016746 177674          MOV      PDRS,-(SP)          ;GET NUMBER OF KIPARS
3670 022062 006316          ASL      (SP)                ;DOUBLE IT
3671 022064 062760 000040 172344          1$:      ADD      #40,KIPAR2(RO) ;INCREMENT KIPAR BY 1K
3672 022072 005720          TST      (RO)+                ;BUMP INDEX
3673 022074 020016          CMP      RO,(SP)              ;DONE?
3674 022076 001372          BNE      1$                  ;BR IF NO
3675 022100 005726          TST      (SP)+                ;CLEAN UP STACK
3676 022102 000207          RTS      PC                  ;RETURN
3677
3678          ;SUB-ROUTINE TO GENERATE "MEX", "WRDCT", "BUF" USING MEMORY MANAGEMENT
3679
3680 022104 013700 172344          PARREG: MOV      2#KIPAR2,RO    ;GET THE LOW REGISTER
3681 022110 042700 171770          BIC      #171770,RO          ;CLEAR ALL BUT ADDRESS BITS 17&18
3682 022114 006300          ASL      RO                  ;PROPERLY POSITION MEX BITS
3683 022116 006300          ASL      RO
3684 022120 000300          SWAB    RO
3685 022122 010067 177624          MOV      RO,MEX              ;STORE FOR DRIVE EXTENDED MEMORY BITS
3686 022126 013700 172344          MOV      2#KIPAR2,RO        ;GET THE LOW REGISTER AGAIN

```

```

3697 022132 042700 006000      BIC      #6000,RO      ;CLEAR ADDRESS BITS 17&18
3698 022136 006200              ASR      RO          ;PROPERLY POSITION AS RPBA
3699 022140 006200              ASR      RO
3690 022142 000300              SWAB     RO
3691 022144 010067 003504      MOV      RO,BUF      ;STORE AS BUS ADDRESS
3692 022150 042767 001000 003454 BIC      #B9,FLAG    ;CLEAR LAST PAGE FLAG
3693 022156 005000              CLR      RO          ;INITIALIZE INDEX
3694 022160 016700 177572      MOV      PDRS,RO     ;GET NUMBER OF KIPARS
3695 022164 005300              DEC      RO
3696 022166 006300              ASL      RO          ;FINAL INDEX VALUE
3697 022170 016046 172344      MOV      KIPAR2(RO),-(SP) ;PUT VALUE OF LAST KIPAR ON STACK
3698 022174 022667 177560      CMP      (SP)+,PARMAX ;WORKING WITH LAST PAGE?
3699 022200 001003              BNE     1$          ;BR IF NOT LAST PAGE
3700 022202 052767 001000 003422 BIS      #B9,FLAG    ;SET LAST PAGE BIT
3701 022210 032767 000002 003414 1$: BIT      #B1,FLAG    ;OPERATOR SELECTING TRANSFER SIZE
3702 022216 001010              BNE     2$          ;BR IF OPERATOR SELECTING TRANSFER SIZE
3703 022220 032767 001000 003404 BIT      #B9,FLAG    ;LAST TRANSFER?
3704 022226 001004              BNE     2$          ;BR IF YES
3705 022230 016767 177526 003404 MOV      MAXWC,WRDCT ;SETUP WORD COUNT
3706 022236 000403              BR      3$          ;EXIT FROM SUB-ROUTINE
3707 022240 016767 003412 003374 2$: MOV      SWRDCT,WRDCT ;SETUP OPERATOR OR LAST PAGE WORD COUNT
3708 022246 000207              3$: RTS      PC      ;RETURN

```

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

```

3714 022250 010667 000222      NPR: MOV      SP,BCKFLG ;SET BACKGROUND FLAG
3715 022254 012767 030000 000210 MOV      #30000,NPRCNT ;SETUP TIMEOUT COUNTER
3716 022262 012701 022474      MOV      #NPR1,R1
3717 022266 005011              CLR      (R1)
3718 022270 000261              SEC
3719 022272              2$:
3720 022272 106111              ROLB     (R1)
3721 022274 105421              NEGB    (R1)+
3722 022276 105441              NEGB    -(R1)
3723 022300 106111              ROLB     (R1)
3724 022302 105421              NEGB    (R1)+
3725 022304 105441              NEGB    -(R1)
3726 022306 106111              ROLB     (R1)
3727 022310 105421              NEGB    (R1)+
3728 022312 105441              NEGB    -(R1)
3729 022314 106111              ROLB     (R1)
3730 022316 105421              NEGB    (R1)+
3731 022320 105441              NEGB    -(R1)
3732 022322 106111              ROLB     (R1)
3733 022324 105421              NEGB    (R1)+
3734 022326 105441              NEGB    -(R1)
3735 022330 106111              ROLB     (R1)
3736 022332 105421              NEGB    (R1)+
3737 022334 105441              NEGB    -(R1)
3738 022336 106111              ROLB     (R1)
3739 022340 105421              NEGB    (R1)+
3740 022342 105441              NEGB    -(R1)
3741 022344 106111              ROLB     (R1)
3742 022346 105421              NEGB    (R1)+

```

3743	022350	105441		NEGB	-(R1)	
3744	022352	005201		INC	R1	
3745	022354	106111		ROLB	(R1)	
3746	022356	105421		NEGB	(R1)+	
3747	022360	105441		NEGB	-(R1)	
3748	022362	106111		ROLB	(R1)	
3749	022364	105421		NEGB	(R1)+	
3750	022366	105441		NEGB	-(R1)	
3751	022370	106111		ROLB	(R1)	
3752	022372	105421		NEGB	(R1)+	
3753	022374	105441		NEGB	-(R1)	
3754	022376	106111		ROLB	(R1)	
3755	022400	105421		NEGB	(R1)+	
3756	022402	105441		NEGB	-(R1)	
3757	022404	106111		ROLB	(R1)	
3758	022406	105421		NEGB	(R1)+	
3759	022410	105441		NEGB	-(R1)	
3760	022412	106111		ROLB	(R1)	
3761	022414	105421		NEGB	(R1)+	
3762	022416	105441		NEGB	-(R1)	
3763	022420	106111		ROLB	(R1)	
3764	022422	105421		NEGB	(R1)+	
3765	022424	105441		NEGB	-(R1)	
3766	022426	106111		ROLB	(R1)	
3767	022430	105421		NEGB	(R1)+	
3768	022432	105441		NEGB	-(R1)	
3769	022434	106111		ROLB	(R1)	
3770	022436	005301		DEC	R1	
3771	022440	103401		BCS	1\$	
3772	022442	000000		HALT		; ARITHMETIC OPERATION FAILED RUN DIAG
3773	022444	005367	000022	DEC	NPRCNT	
3774	022450	001310		BNE	2\$	
3775	022452	104400		HLT		; OPERATION TIMED OUT WAITING FOR INTERRUPT
3776	022454	004567	156604	JSR	R5,PRINT\$	; PRINT MESSAGE
3777	022460	024444		TIMO		
3778	022462	000000		HALT		
3779						
3780	022464	005067	000006	NPRRET: CLR	BCKFLG	
3781	022470	000207		RTS	PC	
3782	022472	000000		NPRCNT: 0		
3783	022474	000000		NPR1: 0		
3784	022476	000000		BCKFLG: 0		
3785						
3786						
3787						
3788						
3789						
3790						
3791	022500	000005		CYLSK: RESET		
3792	022502	004567	156574	JSR	R5,PRNTF\$	; FORCE PRINT THE MESSAGE
3793	022506	025302		CON17		
3794	022510	004767	157170	JSR	PC,READ\$	; INPUT MESSAGE
3795	022514	022767	000131 157272	CMP	#131,INPUT\$	; DOES HE WANT HEAD ALIGNMENT ROUTINE
3796	022522	001002		BNE	4\$	; BR IF NO
3797	022524	000167	000424	JMP	HEAD	; GO DO HEAD ALIGNMENT ROUTINE
3798	022530					

; THIS TEST ALLOWS THE OPERATOR TO SPECIFY TWO CYLINDER ADDRESSES  
; AND THE PROGRAM WILL THEN SEEK BETWEEN THEM. THE ROUTINE DOES  
; NOT CHECK FOR ERRORS.

4\$:

3799	022530	004567	156546		JSR	RS,PRNTFS	;FORCE PRINT THE MESSAGE
3800	022534	025534			CON19		
3801	022536	004767	157142		JSR	PC,READS	;INPUT MESSAGE
3802	022542	022767	000131	157244	CMP	#131,INPUTS	;DOES HE WANT HOME SEEKS?
3803	022550	001002			BNE	1S	;BR IF NO
3804	022552	000167	000516		JMP	HOMERS	;GO DO HOME SEEK ROUTINE
3805	022556						
3806	022556	004567	156520		1S:	JSR	RS,PRNTFS ;FORCE PRINT THE MESSAGE
3807	022562	025254			CON15		;CYLINDER A
3808	022564	004767	157114		JSR	PC,READS	;INPUT MESSAGE
3809	022570	004767	157266		JSR	PC,PACKS	;CONVERT INPUT TO A NUMBER
3810	022574	022767	000626	157532	CMP	#626,NUMS	;IS CYL ADDR TOO HIGH?
3811	022602	101765			BLOS	1S	;BRANCH IF YES
3812	022604	016767	157524	003102	MOV	NUMS,CYLA	;SAVE FIRST ADDR
3813	022612						
3814	022612	004567	156464		2S:	JSR	RS,PRNTFS ;FORCE PRINT THE MESSAGE
3815	022616	025267			CON16		;CYLINDER B
3816	022620	004767	157060		JSR	PC,READS	;INPUT MESSAGE
3817	022624	004767	157232		JSR	PC,PACKS	;CONVERT INPUT TO A NUMBER
3818	022630	022767	000626	157476	CMP	#626,NUMS	;IS CYL ADDR TOO HIGH?
3819	022636	101765			BLOS	2S	;BRANCH IF YES
3820	022640	016767	157470	003050	MOV	NUMS,CYLB	;SAVE SECOND ADDR
3821	022646						
3822	022646	004567	156430		3S:	JSR	RS,PRNTFS ;FORCE PRINT THE MESSAGE
3823	022652	024661			CON4		;DRIVE?
3824	022654	004767	157024		JSR	PC,READS	;INPUT MESSAGE
3825	022660	004767	157176		JSR	PC,PACKS	;CONVERT INPUT TO A NUMBER
3826	022664	022767	000010	157442	CMP	#10,NUMS	;IS UNIT # TOO HIGH?
3827	022672	101765			BLOS	3S	;BRANCH IF YES
3828	022674	000241			CLC		
3829	022676	006167	157432		ROL	NUMS	
3830	022702	006167	157426		ROL	NUMS	
3831	022706	016767	157422	002716	MOV	NUMS,FLAG	;SAVE UNIT NO.
3832	022714	004767	174056		JSR	PC,GATTN	;DETERMINE ATTENTION BIT
3833	022720	004567	173644		JSR	RS,DSKNOS	;SELECT THE UNIT
3834	022724	016777	002764	002664	MOV	CYLA,DRPCA	;LOAD THE CYLINDER ADDR
3835	022732	052777	000011	002646	BIS	#11,DRPCS	;ISSUE SEEK COMMAND
3836	022740	105777	002642		20S:	DRPCS	;WAIT FOR READY
3837	022744	100375			BPL	20S	
3838	022746	036777	174054	002652	1S:	BIT	ATTN,DRPDS ;WAIT FOR ATTENTION
3839	022754	001774			BEQ	1S	
3840	022756	016777	174044	002642	MOV	ATTN,DRPDS	;CLEAR ATTENTION BIT
3841	022764	005777	002616		TST	DRPCS	;ANY ERRORS?
3842	022770	100022			BPL	2S	
3843	022772	104400			HLT		;ERROR AFTER SEEK COMMAND
3844	022774	032777	004000	002624	BIT	#B11,DRPDS	;SEEK INCOMPLETE?
3845	023002	001415			BEQ	2S	;BRANCH IF NO
3846	023004	112777	000015	002574	MOV	#15,DRPCS	;ISSUE HOME COMMAND
3847	023012	105777	002570		21S:	DRPCS	;WAIT FOR READY
3848	023016	100375			BPL	21S	
3849	023020	036777	174002	002600	3S:	BIT	ATTN,DRPDS ;WAIT FOR ATTENTION BIT
3850	023026	001774			BEQ	3S	
3851	023030	016777	173772	002570	MOV	ATTN,DRPDS	
3852	023036	016777	002654	002552	2S:	MOV	CYLB,DRPCA ;LOAD CYLINDER ADDR
3853	023044	052777	000011	002534	BIS	#11,DRPCS	;ISSUE SEEK COMMAND
3854	023052	105777	002530		22S:	TSTB	DRPCS ;WAIT FOR READY



```

3855 023056 100375          BPL      22$
3856 023060 036777 173742 002540 4$: BIT      ATTN,ARPD$ ;WAIT FOR ATTENTION
3857 023066 001774          BEQ      4$
3858 023070 016777 173732 002530 MOV      ATTN,ARPD$ ;CLEAR ATTENTION
3859 023076 005777 002504 TST      ARPCS ;ANY ERRORS?
3860 023102 100022          BPL      5$
3861 023104 104400          HLT
3862 023106 032777 004000 002512 BIT      #811,ARPD$ ;SEEK INCOMPLETE?
3863 023114 001415          BEQ      5$
3864 023116 112777 000015 002462 MOVB     #15,ARPCS ;ISSUE HOME COMMAND
3865 023124 105777 002456 23$: TSTB     ARPCS ;WAIT FOR READY
3866 023130 100375          BPL      23$
3867 023132 036777 173670 002466 6$: BIT      ATTN,ARPD$ ;WAIT FOR ATTENTION
3868 023140 001774          BEQ      6$
3869 023142 016777 173660 002456 MOV      ATTN,ARPD$ ;CLEAR ATTENTION
3870 023150 000167 177544 5$: JMP      CYLS1
3871
3872 ;THIS ROUTINE IS USED FOR HEAD ALIGNMENT - SWITCHES 00-04 IS THE
3873 ;HEAD TO BE SELECTED, SWITCHES 08-10 IS THE DRIVE TO BE SELECTED -
3874 ;WHEN SWITCH 07 IS TOGGLED A NEW HEAD AND DRIVE WILL BE SELECTED.
3875
3876 023154 000005          HEAD: RESET
3877 023156 004567 156120 JSR      RS,PRNTF$ ;FORCE PRINT THE MESSAGE
3878 023162 025347          CON18
3879 023164 000000          HALT
3880 023166 005067 002214 CLR      CON18A ;CLEAR OUT SWITCH DEFINATIONS PRINTOUT
3881 023172 012777 000222 002416 1$: MOV      #222,ARPCA ;SETUP FOR CYLINDER 146
3882 023200 013746 177570 MOV      @#SWR,-(SP)
3883 023204 011667 002472 MOV      (SP),WORK
3884 023210 042767 177600 002464 BIC      #177600,WORK ;CLEAR OUT DRIVE NUMBER
3885 023216 000367 002460 SWAB     WORK ;PUT HEAD IN PROPER POSITION
3886 023222 016777 002454 002370 MOV      WORK,ARPD$ ;LOAD HEAD NUMBER INTO DISK ADDRESS
3887 023230 012667 002446 MOV      (SP)+,WORK ;GET DRIVE NUMBER
3888 023234 042767 000377 002440 BIC      #377,WORK ;GE RID OF HEAD ADDRESS
3889 023242 052767 000011 002432 BIS      #11,WORK ;SET A SEEK GO TO DRIVE NUMBER
3890 023250 016777 002426 002330 MOV      WORK,ARPCS ;SET SEEK
3891 023256 005777 002344 TST      ARPD$
3892 023262 100375          BPL      -4 ;WAIT FOR DRIVE TO BE READY
3893 023264 105737 177570 TSTB     @#SWR ;NEW HEAD AND/OR DRIVE
3894 023270 100375          BPL      -4 ;BR IF NO
3895 023272 000737          BR      1$ ;GO TO NEW DRIVE AND HEAD
3896
3897
3898 ;THIS ROUTINE DOES A HOME SEEK AND THEN TRIES TO DO
3899 ;A TWO WORD READ FROM CYLINDER 000 WITH OUT A SEEK TO CYLINDER 000
3900 ;LOOPING BACK TO HOME SEEK
3901
3902 HOMERS:
3903 023274 004567 156002 JSR      RS,PRNTF$ ;FORCE PRINT THE MESSAGE
3904 023300 024661          CON4 ;WHAT DRIVE?
3905 023302 004767 156376 JSR      PC,READ$ ;INPUT MESSAGE
3906 023306 004767 156550 JSR      PC,PACK$ ;CONVERT INPUT TO A NUMBER
3907 023312 022767 000010 157014 CMP      #10,NUM$ ;TOO LARGE?
3908 023320 101765          BLOS     HOMERS ;BR IF YES
3909 023322 116777 157006 002260 MOVB     NUM$,ARPCS1 ;LOAD UNIT NUMBER INTO CONTROLER
3910 023330 112777 000015 002250 1$: MOVB     #15,ARPCS ;ISSUE A HOME SEEK AND GO

```



```

3911 023336 105777 002244 2$: TSTB 2RPCS ;CONTROLLER READY?
3912 023342 100375 3PL 2$ ;BR IF NO
3913 023344 005777 002256 3$: TST 2RPDS ;DRIVE READY?
3914 023350 100375 BPL 3$ ;BR IF NO
3915 023352 005777 002230 TST 2RPCS ;ERRORS?
3916 023356 100012 BPL 4$ ;BR IF NO ERRORS
3917 023360 104400 HLT ;ERROR WITH HOME SEEKS
3918 023362 032777 001000 002236 BIT #89,2RPDS ;FILE UNSAFE?
3919 023370 001401 BEQ 5$ ;BR IF NO
3920 023372 000000 HALT ;FILE UNSAFE
3921 023374 012767 023330 155400 5$: MOV #1$,LAD ;SETUP SCOPE LOOP
3922 023402 104000 SCOPE
3923 023404 005077 002206 4$: CLR 2RPCA ;SETUP FOR CYLINDER 000
3924 023410 005077 002204 CLR 2RPDA ;SETUP FOR TRACK 00, SECTOR 00
3925 023414 012777 177776 002170 MOV #-2,2RPWC ;SETUP 2 WORD TRANSFER
3926 023422 012777 025730 002164 MOV #OUTBUF,2RPBA ;INTO OUTPUT BUFFER
3927 023430 052777 000017 002150 BIS #17,2RPCS ;READ WITH NO IMPLIED SEEK
3928 023436 105777 002144 6$: TSTB 2RPCS ;DONE?
3929 023442 100375 BPL 6$ ;BR IF NO
3930 023444 005777 002136 TST 2RPCS ;ERRORS?
3931 023450 100327 BPL 1$ ;BR IF NO ERRORS TO MORE HOME SEEKS
3932 023452 104400 HLT ;HOME SEEK DID NOT RETURN TO CYLINDER 000
3933 ;FOR A READ OF ONE WORD
3934 023454 012767 023330 155320 MOV #1$,LAD ;SETUP FOR SCOPE LOOP
3935 023462 104000 SCOPE
3936 023464 000703 BR HOMERS ;TRY TEST AGAIN WITH NEW DRIVE
  
```

:THIS ROUTINE RESTORES THE LOADERS FROM BK TO HIGHEST NON MEMORY  
 ;MANAGEMENT CORE

```

3944 023466 012700 034000 RELOAD: MOV #34000,R0 ;START OF LOADERS
3945 023472 016701 176252 MOV MEMSIZ,R1 ;TOP OF MEMORY
3946 023476 042701 003777 BIC #3777,R1 ;MAKE IT A 1K TRANSFER
3947 023502 012737 023522 000004 MOV #1$,2#ERRVEC ;SET UP FOR TRAP
3948 023510 012737 000340 000006 MOV #340,2#ERRVEC+2
3949 023516 012021 2$: MOV (R0)+,(R1)+ ;RESTORE LOADER
3950 023520 000776 BR 2$
3951 023522 012737 000006 000004 1$: MOV #6,2#ERRVEC ;RESTORE TRAPCATCHER
3952 023530 005037 000006 CLR 2#ERRVEC+2
3953 023534 005037 000176 CLR 2#176
3954 023540 005067 002160 CLR LDRFLG ;SET LOADER FLAG TO LOADERS AT TOP OF MEMORY
3955 023544 012707 000176 MOV #176,PC ;FINISHED
  
```

;ERROR MESSAGE HEADERS

```

3959 .EVEN
3960 023550 005015 047105 020104 MES1: .ASCIZ <15><12>/END OF PASS /
3961 023556 043117 050040 051501
3962 023564 020123 000
3963 023567 015 051012 042520 MES1A: .ASCIZ <15><12>/RPER= /
3964 023574 036522 000040
3965
3966 023600 005015 050122 051503 MES2: .ASCIZ <15><12>/RPCS= /
  
```

3967	023606	020075	000						
3968									
3969	023611	015	051012	042120	MES2A:	.ASCIZ	<15><12>/RPDS=	/	
3970	023616	036523	000040						
3971	023622	005015	050122	040503	MES2B:	.ASCIZ	<15><12>/RPCA=	/	
3972	023630	020075	000						
3973	023633	015	051012	042120	MES2C:	.ASCIZ	<15><12>/RPDA=	/	
3974	023640	036501	000040						
3975	023644	005015	052523	040503	MES2D:	.ASCIZ	<15><12>/SUCA=	/	
3976	023652	020075	000						
3977									
3978	023655	015	046412	046505	MES4:	.ASCIZ	<15><12>/MEM ADDR=	/	
3979	023662	040440	042104	036522					
3980	023670	000040							
3981									
3982	023672	005015	040527	052111	MES5:	.ASCIZ	<15><12>/WAIT 5 SEC. AND TURN OFF PDP-11 PWR/		
3983	023700	032440	051440	041505					
3984	023706	020056	047101	020104					
3985	023714	052524	047122	047440					
3986	023722	043106	050040	050104					
3987	023730	030455	020061	053520					
3988	023736	000122							
3989									
3990	023740	005015	042524	052123	MES6:	.ASCIZ	<15><12>/TEST NO	/	
3991	023746	047040	020117	000					
3992									
3993	023753	015	052012	052117	MES7:	.ASCIZ	<15><12>/TOT REREADS ON ERR=	/	
3994	023760	051040	051105	040505					
3995	023766	051504	047440	020116					
3996	023774	051105	036522	000040					
3997									
3998									
3999	024002	005015	052123	052101	MES8:	.ASCIZ	<15><12>/STAT ERR/		
4000	024010	042440	051122	000					
4001									
4002	024015	015	053412	020104	MES9:	.ASCIZ	<15><12>/WD CNT INTO SECT=	/	
4003	024022	047103	020124	047111					
4004	024030	047524	051440	041505					
4005	024036	036524	000040						
4006									
4007	024042	005015	040520	020124	MES10:	.ASCIZ	<15><12>/PAT IN USE=	/	
4008	024050	047111	052440	042523					
4009	024056	020075	000						
4010									
4011	024061	015	052412	044516	MES11:	.ASCIZ	<15><12>/UNIT NO.	/	
4012	024066	020124	047516	020056					
4013	024074	000							
4014									
4015	024075	103	046131	000	MES12:	.ASCIZ	/CYL/		
4016									
4017	024101	015	051012	040505	MES13:	.ASCIZ	<15><12>/READ NO.	/	
4018	024106	020104	047516	020056					
4019	024114	000							
4020									
4021	024115	015	041412	046131	MES14:	.ASCIZ	<15><12>/CYL=	/	
4022	024122	020075	000						

```

4023
4024 024125 015 044012 040505 MES15: .ASCIZ <15><12>/HEAD= /
4025 024132 036504 000040
4026
4027 024136 005015 042523 052103 MES16: .ASCIZ <15><12>/SECT= /
4028 024144 020075 000
4029
4030 024147 015 041412 046517 MES17: .ASCIZ <15><12>/COMP ERR/
4031 024154 020120 051105 000122
4032
4033 024162 005015 054105 042520 MES18: .ASCIZ <15><12>/EXPECTED /
4034 024170 052103 042105 000040
4035
4036 024176 005015 042522 053103 MES19: .ASCIZ <15><12>/RECVD /
4037 024204 020104 000
4038 024207 015 047012 020117 MES20: .ASCIZ <15><12>/NO UNITS AVAIL/
4039 024214 047125 052111 020123
4040 024222 053101 044501 000114
4041 024230 005015 042522 042101 MES21: .ASCIZ <15><12>/READ CNTR = /
4042 024236 041440 052116 020122
4043 024244 020075 000040
4044 024250 005015 042114 051522 MES22: .ASCII <15><12>/LDRS MOVED TO BK/
4045 024256 046440 053117 042105
4046 024264 052040 020117 045470
4047 024272 005015 047524 051040 .ASCIZ <15><12>/TO REST LDR START AT LOC /
4048 024300 051505 020124 042114
4049 024306 020122 052123 051101
4050 024314 020124 052101 046040
4051 024322 041517 020040 000
4052 024327 015 045412 050111 MES23: .ASCIZ <15><12>/KIPAR2 = /
4053 024334 051101 020062 020075
4054 024342 000040
4055 024344 020040 020040 020040 MES24: .ASCIZ / KIPAR3 = /
4056 024352 020040 044513 040520
4057 024360 031522 036440 020040
4058 024366 000
4059 024367 015 045412 050111 MES25: .ASCIZ <15><12>/KIPAR4 = /
4060 024374 051101 020064 020075
4061 024402 000040
4062 024404 020040 020040 020040 MES26: .ASCIZ / KIPAR5 = /
4063 024412 020040 044513 040520
4064 024420 032522 036440 020040
4065 024426 000
4066 024427 015 045412 050111 MES27: .ASCIZ <15><12>/KIPAR6 = /
4067 024434 051101 020066 020075
4068 024442 000040
4069
4070 024444 005015 051120 041517 TIMO: .ASCIZ <15><12>/PROC BCKGRD TEST TIMED OUT/
4071 024452 041040 045503 051107
4072 024460 020104 042524 052123
4073 024466 052040 046511 042105
4074 024474 047440 052125 000
4075
4076 ;CONVERSATION TEXT
4077 ;
4078 ;

```

4079	024501	015	051412	042124	SPECMES:	.ASCIZ <15><12>/STDARD WDS XFERRED= /
4080	024506	051101	020104	042127		
4081	024514	020123	043130	051105		
4082	024522	042522	036504	000040		
4083						
4084	024530	005015	040504	040524	CON1:	.ASCIZ <15><12>/DATA TEST ONLY? (Y OR N)/
4085	024536	052040	051505	020124		
4086	024544	047117	054514	020077		
4087	024552	054450	030040	020122		
4088	024560	024516	000			
4089						
4090	024563	015	046412	046125	CON2:	.ASCIZ <15><12>/MULTI DRIVE MODE?(Y OR N)/
4091	024570	044524	042040	044522		
4092	024576	042526	046440	042117		
4093	024604	037505	054450	047440		
4094	024612	020122	024516	000		
4095						
4096	024617	015	047012	046525	CON3:	.ASCIZ <15><12>/NUMBER OF DRIVES 1 TO 10 OCTAL?/
4097	024624	042502	020122	043117		
4098	024632	042040	044522	042526		
4099	024640	020123	020061	047524		
4100	024646	030440	020060	041517		
4101	024654	040524	037514	000		
4102						
4103	024661	015	053412	044510	CON4:	.ASCIZ <15><12>/WHICH DRIVE?/
4104	024666	044103	042040	044522		
4105	024674	042526	000077			
4106						
4107	024700	005015	050117	044524	CON5:	.ASCIZ <15><12>/OPTIONAL WORD COUNT? (Y OR N)/
4108	024706	047117	046101	053440		
4109	024714	051117	020104	047503		
4110	024722	047125	037524	024040		
4111	024730	020131	051117	047040		
4112	024736	000051				
4113						
4114	024740	005015	042514	043516	CON6:	.ASCIZ <15><12>/LENGTH? (1 TO SWRDCT)/
4115	024746	044124	020077	030450		
4116	024754	052040	020117	053523		
4117	024762	042122	052103	000051		
4118						
4119	024770	005015	052123	051101	CON7:	.ASCIZ <15><12>/STARTING HEAD?/
4120	024776	044524	043516	044040		
4121	025004	040505	037504	000		
4122						
4123	025011	015	042012	020117	CON7A:	.ASCIZ <15><12>/DO YOU WISH TO SELECT THE DISK TEST ADDR?(Y OR N)/
4124	025016	047531	020125	044527		
4125	025024	044123	052040	020117		
4126	025032	042523	042514	052103		
4127	025040	052040	042510	042040		
4128	025046	051511	020113	042524		
4129	025054	052123	040440	042104		
4130	025062	037522	054450	047440		
4131	025070	020122	024516	000		
4132						
4133	025075	015	051412	040524	CON7B:	.ASCIZ <15><12>/STARTING SECT?/
4134	025102	052122	047111	020107		

4135	025110	042523	052103	000077	
4136					
4137	025116	005015	052123	051101	CON7C: .ASCIZ <15><12>/STARTING CYL?/
4138	025124	044524	043516	041440	
4139	025132	046131	000077		
4140					
4141	025136	005015	050117	040524	CON8: .ASCIZ <15><12>/OPTAL DATA PAN NO.?/
4142	025144	020114	040504	040524	
4143	025152	050040	047101	047040	
4144	025160	027117	000077		
4145					
4146	025164	005015	051127	052111	CON9: .ASCIZ <15><12>/WRITE?(Y OR N)/
4147	025172	037505	054450	047440	
4148	025200	020122	024516	000	
4149					
4150	025205	015	053412	044522	CON10: .ASCIZ <15><12>/WRITE CHECK?(Y OR N)/
4151	025212	042524	041440	042510	
4152	025220	045503	024077	020131	
4153	025226	051117	047040	000051	
4154					
4155	025234	005015	042522	042101	CON11: .ASCIZ <15><12>/READ?(Y OR N)/
4156	025242	024077	020131	051117	
4157	025250	047040	000051		
4158					
4159	025254	005015	054503	020114	CON15: .ASCIZ <15><12>/CYL "A"??/
4160	025262	040442	037442	000	
4161					
4162	025267	015	041412	046131	CON16: .ASCIZ <15><12>/CYL "B"??/
4163	025274	021040	021102	000077	
4164	025302	005015	040527	052116	CON17: .ASCIZ <15><12>/WANT HEAD ALIGN ROUTINE?(Y OR N) /
4165	025310	044040	040505	020104	
4166	025316	046101	043511	020116	
4167	025324	047522	052125	047111	
4168	025332	037505	054450	047440	
4169	025340	020122	024516	020040	
4170	025346	000			
4171	025347	015	051412	052105	CON18: .ASCII <15><12>/SET SWCH THEN PRESS CONTINUE./
4172	025354	051440	041527	020110	
4173	025362	044124	047105	050040	
4174	025370	042522	051523	041440	
4175	025376	047117	044524	052516	
4176	025404	027105			
4177					
4178	025406	005015	053523	044103	.EVEN CON18A: .ASCII <15><12>/SWCH 00-04 = HEAD/
4179	025414	030040	026460	032060	
4180	025422	036440	044040	040505	
4181	025430	104			
4182	025431	015	051412	041527	.ASCII <15><12>/SWCH 08-10 = DRIVE/
4183	025436	020110	034060	030455	
4184	025444	020060	020075	051104	
4185	025452	053111	105		
4186	025455	015	052012	043517	.ASCIZ <15><12>'TOGGLE SW 07 TO SELECT NEW HEAD AND/OR DRIVE'
4187	025462	046107	020105	053523	
4188	025470	030040	020067	047524	
4189	025476	051440	046105	041505	
4190	025504	020124	042516	020127	

4191	025512	042510	042101	040440	
4192	025520	042116	047457	020122	
4193	025526	051104	053111	000105	
4194	025534	005015	040527	052116	CON19: .ASCIZ <15><12>/WANT CONTINUOUS HOME SEEKS? /
4195	025542	041440	047117	044524	
4196	025550	052516	052517	020123	
4197	025556	047510	042515	051440	
4198	025564	042505	051513	020077	
4199	025572	000			
4200					
4201	025573	015	042412	042116	END: .ASCIZ <15><12>/END/
4202	025600	000			
4203		025602			.EVEN

4204  
4205  
4206  
4207  
4208  
4209  
4210  
4211  
4212  
4213  
4214  
4215  
4216  
4217  
4218  
4219  
4220  
4221  
4222  
4223  
4224  
4225  
4226  
4227  
4228  
4229  
4230  
4231  
4232  
4233  
4234  
4235  
4236  
4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259

025602 000254  
025604 000256  
025606 176714  
025610 176715  
025612 176716  
025614 176720  
025616 176722  
025620 176724  
025622 176725  
025624 176712  
025626 176710  
025630 176734  
  
025632 000000  
025634 000000  
025636 000000  
025640 000000  
025642 000000  
025644 000000  
025646 000000  
025650 000000  
025652 000000  
025654 000000  
025656 000000  
025660 000000  
025662 000000  
025664 000000  
025666 000000  
025670 000000  
025672 000000  
025674 000000  
025676 000000  
025700 000000  
025702 000000  
025704 000000  
025706 000000  
025710 000000  
025712 000000  
025714 000000  
025716 000000  
025720 000000  
025722 000000  
025724 000000

;  
;  
;  
;  
;  
;  
;  
;DISK I/O REGISTERS

VECTOR: 254  
STATUS: 256  
RPCS: 176714  
RPCS1: 176715  
RPWC: 176716  
RPBA: 176720  
RPCA: 176722  
RPDA: 176724  
RPDA1: 176725  
RPER: 176712  
RPDS: 176710  
SUCA: 176734

;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

;DEDICATED REGISTERS

FLAG: 0  
SCYL: 0  
SHED: 0  
SSEC: 0  
WRDCT: 0  
CYLINDER: 0  
DMA: 0  
SWITCH: 0  
PATNU: 0  
BUF: 0  
SWRDC: 0  
SAVE: 0  
DSKNOR: 0  
BLOCK: 0  
PASSC: 0  
INBUF: 0  
RDERR: 0  
ACNVX: 0  
TESTNO: 0  
ERCOUNT: 0  
WORK: 0  
WORK1: 0  
WORK2: 0  
WORK3: 0  
WORK4: 0  
CYLA: 0  
CYLB: 0  
SEEK1: 0  
CNTA: 0  
LDRFLG: 0

;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  
  
;CONTAINS START OF INPUT BUFFER  
;READ RETRY COUNTER  
  
;WHEN =0 LOADERS ARE AT TOP OF MEMORY

.EVEN

4260	025726	005015			HEADER: .ASCII <15><12>
4261	025730	042115	030455	026461	OUTBUF: .ASCII 'MD-11-DZRPB-C DISK RELIABILITY TEST'
4262	025736	055104	050122	026502	
4263	025744	020103	020040	044504	
4264	025752	045523	051040	046105	
4265	025760	040511	044502	044514	
4266	025766	054524	052040	051505	
4267	025774	124			
4268	025775	015	005012	053523	.ASCII <15><12><12>'SWITCH OPTIONS'
4269	026002	052111	044103	047440	
4270	026010	052120	047511	051516	
4271	026016	005015	053523	036040	.ASCII <15><12>'SW <15> =1....HALT ON ERROR'
4272	026024	032461	020076	030475	
4273	026032	027056	027056	040510	
4274	026040	052114	047440	020116	
4275	026046	051105	047522	122	
4276	026053	015	051412	020127	.ASCII <15><12>'SW <14> =1....LOOP ON ERROR'
4277	026060	030474	037064	036440	
4278	026066	027061	027056	046056	
4279	026074	047517	020120	047117	
4280	026102	042440	051122	051117	
4281	026110	005015	053523	036040	.ASCII <15><12>'SW <13> =1....INHIBIT PRINTOUTS'
4282	026116	031461	020076	030475	
4283	026124	027056	027056	047111	
4284	026132	044510	044502	020124	
4285	026140	051120	047111	047524	
4286	026146	052125	123		
4287	026151	015	051412	020127	.ASCII <15><12>'SW <12> =1....INHIBIT BACKGROUND TEST'
4288	026156	030474	037062	036440	
4289	026164	027061	027056	044456	
4290	026172	044116	041111	052111	
4291	026200	041040	041501	051113	
4292	026206	052517	042116	052040	
4293	026214	051505	124		
4294	026217	015	051412	020127	.ASCII <15><12>'SW <11> =1....RING BELL ON ERROR'
4295	026224	030474	037061	036440	
4296	026232	027061	027056	051056	
4297	026240	047111	020107	042502	
4298	026246	046114	047440	020116	
4299	026254	051105	047522	122	
4300	026261	015	051412	020127	.ASCII <15><12>'SW <10> =1....LOOP ON TEST'
4301	026266	030474	037060	036440	
4302	026274	027061	027056	046056	
4303	026302	047517	020120	047117	
4304	026310	052040	051505	124	
4305	026315	015	051412	020127	.ASCII <15><12>'SW <09> =1....INHIBIT DATA COMPARISION'
4306	026322	030074	037071	036440	
4307	026330	027061	027056	044456	
4308	026336	044116	041111	052111	
4309	026344	042040	052101	020101	
4310	026352	047503	050115	051101	
4311	026360	051511	047511	116	
4312	026365	015	051412	020127	.ASCII <15><12>'SW <08> =1....ENTER CONVERSATION MODE'
4313	026372	030074	037070	036440	
4314	026400	027061	027056	042456	
4315	026406	052116	051105	041440	



4316	026414	047117	042526	051522
4317	026422	052101	047511	020116
4318	026430	047515	042504	
4319	026434	005015	020040	020040
4320	026442	020040	020040	020040
4321	026450	020040	020040	042524
4322	026456	052123	051440	046105
4323	026464	041505	042524	020104
4324	026472	020060	020055	020062
4325	026500	044507	042526	020123
4326	026506	042101	051104	051505
4327	026514	020123	047503	053116
4328	026522	051105	040523	044524
4329	026530	047117		
4330	026532	005015	020040	020040
4331	026540	020040	020040	020040
4332	026546	020040	020040	042524
4333	026554	052123	051440	046105
4334	026562	041505	042524	020104
4335	026570	020063	020055	020067
4336	026576	044507	042526	020123
4337	026604	040504	040524	041440
4338	026612	047117	042526	051522
4339	026620	052101	047511	116
4340	026625	015	051412	020127
4341	026632	030074	037067	036440
4342	026640	027061	027056	044456
4343	026646	044116	041111	052111
4344	026654	051040	047101	047504
4345	026662	020115	042523	045505
4346	026670	020123	052504	044522
4347	026676	043516	042040	052101
4348	026704	020101	042524	052123
4349	026712	005015	053523	036040
4350	026720	033060	020076	030475
4351	026726	027056	027056	047111
4352	026734	044510	044502	020124
4353	026742	051525	020105	043117
4354	026750	046440	046505	051117
4355	026756	020131	040515	040516
4356	026764	042507	042515	052116
4357	026772	005015	053523	036040
4358	027000	032460	027076	027056
4359	027006	027056	027056	047503
4360	027014	050115	051101	051511
4361	027022	047511	020116	051105
4362	027030	047522	051522	050040
4363	027036	044522	052116	052517
4364	027044	124		
4365	027045	015	051412	020127
4366	027052	030074	037064	027056
4367	027060	027056	027056	051056
4368	027066	051105	040505	020104
4369	027074	047111	047506	046522
4370	027102	052101	047511	116
4371	027107	015	051412	020127

.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'

.ASCII <15><12>'SW <05>.....COMPARISION ERRORS PRINTOUT'

.ASCII <15><12>'SW <04>.....REREAD INFORMATION'

.ASCIZ <15><12>'SW <03> =1....SELECT TEST IN SWCH 00 TO 02'

4372	027114	030074	037063	036440
4373	027122	027061	027056	051456
4374	027130	046105	041505	020124
4375	027136	042524	052123	044440
4376	027144	020116	053523	044103
4377	027152	030040	020060	047524
4378	027160	030040	000062	
4379				
4380	000001			

.END



CON15	025254	3807	4159#											
CON16	025267	3815	4162#											
CON17	025302	3793	4164#											
CON18	025347	3878	4171#											
CON18A	025406	3880*	4178#											
CON19	025534	3800	4194#											
CON2	024563	1226	4090#											
CON3	024617	1233	4096#											
CON4	024661	1249	3823	3904	4103#									
CON5	024700	1260	4107#											
CON6	024740	1266	4114#											
CON7	024770	1295	4119#											
CON7A	025011	1280	4123#											
CON7B	025075	1303	4133#											
CON7C	025116	1287	4137#											
CON9	025136	1312	4141#											
CON9	025164	1325	4146#											
CYLA	025714	1442*	1443*	1444*	1445*	1527*	1528*	1529*	1530*	2069*	2070*	2071*	2072*	3812*
		3834	4252#											
CYLB	025716	3820*	3852	4253#										
CYLIND	025644	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*	2218*	2271*	2485*	2672*	2789*	2836*	2917	3013	3067*	3099*	3100
		3112*	3119	3448	4232#									
CYLSK	022500	1213	3791#											
CYLSI	022720	3833#	3870											
DATAT	012272	1388	1395	2451	2474#									
DATP	012430	2492	2497#	2625										
DATTES	003004	1229	1247#	1253										
DECBLK	016234	3088#	3097	3102										
DECCY	015416	2959	2963#											
DECTK	015372	2955	2958#											
DELNES	015322	2942	2949#											
DISBUF	016160	1611	1668	2514	2528	2619	3078#							
DISPLA=	177570	822#												
DKINT	015224	2489	2931#	3173										
DKII	015530	2939	2981#											
DMA	025646	1201*	1542*	1548	1551	1553*	1555*	1579	1640	1645	1647*	1651	1658*	1689*
		1777*	1924*	1943*	1944*	1964*	1992*	2024*	2151*	2217*	2270*	2486*	2680*	2798*
		2837*	2916	3068*	3069*	3083	3087*	3091*	3092	3096*	3098*	3111*	3118	3447
		4233#												
DREAD	012616	2517	2530#											
DSKDR	002712	1231#	1237	1240										
DSKNCR	025662	1199*	1241*	1242*	1244*	1245*	1350*	1352	1355*	1357*	1359*	1360*	1362	2794
		4239#												
DSKNOS	016570	1416	1514	1688	1775	2116	2150	2829	2915	3005	3154#	3833		
DSKRD	012676	2543#	2589	2595										
EMTVEC=	000030	812#												
END	025573	2895	4201#											
EOPTST	007110	1924#	1982											
ERCOUN	025700	3382*	3492*	4246#										
ERRFLG	001240	931	933*	982*	985#	2583	2745	2931*	2935*	2947*	3006	3429*		
ERROR	001112	962#	3169											
ERRORS	001246	983*	988#											
ERRPC	001252	972	990#											
ERRVEC=	000004	807#	1181*	1182*	1193*	1194*	3551*	3552*	3612*	3616*	3617*	3947*	3948*	3951*











RADT2	004534	1514#	1672											
RADT3	005616	1687#	1756											
RANDOM	017146	2664	3224	3239#										
RANDS	001564	1054#	2666	3405										
RANEX	013366	1396	2635	2643#										
RANLOP	013642	2681#	2693											
RDATAT	012330	2481#	2632	2636										
RDERR	025672	2540*	2585*	2586	2588	2596	2598	2604	2606*	2705*	2747*	2748	2750	2758
		2760	2766	2768*	2936	2945	3430	3488	4243#					
RDSECT	005302	1619#	1666	1669										
READLS	002060	1105	1111#											
READMS	002054	1096	1110#											
READS	001704	1084#	1221	1227	1234	1250	1261	1267	1281	1288	1296	1304	1313	1326
		1333	1340	3794	3801	3808	3816	3824	3905					
READ1	012656	2534	2539#	2622										
RECS	017760	1462*	1571*	1572*	1573	1578*	1586	1590*	1598	1652*	1731*	1898*	1908*	1974*
		1975	2058*	2251*	2307*	2392*	2437*	2877*	2887*	3350	3354#	3428*		
REDAC	015430	2953	2957	2962	2965#									
RELOAD	023466	3636	3944#											
REPOEN	014444	2791	2797	2800#										
RESTS	001064	950#	1079	1155										
RESVEC=	000010	808#												
RMEMT	010604	2201#	2449	2452										
RMMEMT	011060	2203	2259#											
RORBLK	016762	3184	3188#											
RPBA	025614	2918*	3926*	4216#										
RPCA	025616	1422*	1590	2117*	2124*	2830*	2917*	2949	3028*	3295	3834*	3852*	3881*	3923*
		4217#												
RPCS	025606	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	2174*	2221	2223	2234
		2236	2239	2274	2276	2288	2290	2293	2324	2330	2334	2366	2368	2372
		2379	2381	2385	2395	2406	2408	2412	2416	2424	2426	2430	2440	2581
		2590*	2591	2742	2752*	2753	2831*	2840	2846	2848	2857	2864*	2871*	2924*
		2932	2941	2986*	2987	3030*	3031*	3033	3036	3047*	3048	3160*	3291	3508*
		3509	3835*	3836	3841	3846*	3847	3853*	3854	3859	3864*	3865	3890*	3910*
		3911	3915	3927*	3928	3930	4213#							
RPCS1	025610	1352*	3909*	4214#										
RPDA	025620	1571	2118*	2916*	2950	3299	3886*	3924*	4218#					
RPDA1	025622	1578	4219#											
RPDS	025626	1353	1421*	1434	1440	1455	1468	1473	1525	1602	1607	1617	1659	1664
		1706	1711	1734	1739	2078	2122	2137	2142	2175	2593	2755	2842	2872
		2981	2984	2989	3029*	3038	3050	3053	3161	3283	3838	3840*	3844	3849
		3851*	3856	3858*	3862	3867	3869*	3891	3913	3918	4221#			
RPER	025624	1824	1955	2002	2090	2326	2952	3287	4220#					
RPWC	025612	2128*	2919*	2920*	3925*	4215#								
RRANEX	013420	2649#	2780											
RW =	000006	864#	1187	1188	1189	2353	2354	2355	2356	3554	3622	3625	3628	3631
RWRCK	006262	1774#	2179											
RO =%	000000	783#	945	951*	1056*	1060*	1069*	1075*	1077	1119*	1120	1122*	1124*	1158
		1161	1162*	1381*	1382*	1384*	1385	1425*	1426*	1438*	1439*	1446	1516*	1517*
		1523*	1524*	1531	1544*	1546*	1548*	1558*	1559*	1562	1619*	1621*	1634*	1778*
		1807	1838	1844*	1862*	1864	1901*	1903	1908	1996*	1997*	2026*	2027*	2028*









ADC	1066	1068	1070	1071	1073	1076	3251	3253	3255	3256	3258	3261			
ADD	997	1002	1065	1067	1069	1072	1074	1075	1153	2054	2097	2629	2798	2922	3008
	3113	3187	3250	3252	3254	3257	3259	3260	3418	3446	3519	3538	3557	3590	3595
ASL	3598	3604	3663	3671											
ASR	1060	3245	3670	3682	3683	3696									
BCC	1948	3688	3689												
BCS	1850														
BFG	1842	3771													
	930	966	996	1014	1017	1037	1040	1087	1121	1212	1237	1270	1318	1345	1349
	1370	1380	1387	1433	1460	1469	1474	1488	1552	1574	1587	1593	1599	1603	1608
	1632	1646	1660	1665	1671	1703	1707	1712	1729	1735	1740	1755	1808	1893	1996
	1976	2123	2138	2143	2178	2203	2240	2294	2314	2390	2396	2413	2431	2435	2448
	2451	2455	2499	2502	2517	2547	2584	2587	2597	2615	2624	2631	2635	2746	2749
	2759	2775	2779	2791	2806	2849	2875	2937	2939	2955	2959	2982	2985	2992	3039
	3079	3086	3089	3095	3101	3123	3125	3130	3135	3184	3224	3226	3233	3265	3306
	3318	3322	3392	3417	3422	3427	3431	3433	3450	3453	3458	3491	3493	3528	3568
	3570	3572	3589	3621	3624	3627	3630	3652	3655	3658	3661	3839	3845	3850	3857
	3863	3868	3919												
BGE	2599	2761													
BHI	2675	2678	3420												
BIC	1163	1210	1242	1323	1376	1382	1572	2173	2208	2669	2673	2676	2793	2796	2870
	2951	3009	3030	3084	3091	3093	3114	3120	3159	3185	3200	3536	3540	3611	3681
	3687	3692	3884	3888	3946										
BICB	1091	2679													
BIS	982	1132	1138	1143	1151	1192	1206	1224	1230	1257	1277	1284	1318	1329	1336
	1343	1361	1424	1481	1515	1614	1747	1849	2831	2923	2924	2961	3031	3148	3508
	3561	3618	3700	3835	3853	3889	3927								
BISB	1041														
BIT	929	965	969	995	1013	1016	1028	1179	1204	1344	1369	1379	1386	1434	1455
	1468	1473	1487	1602	1607	1631	1659	1664	1670	1706	1711	1734	1739	1754	1824
	1892	1955	1959	2002	2008	2048	2078	2082	2086	2090	2122	2137	2142	2177	2197
	2202	2239	2259	2293	2313	2317	2326	2330	2334	2372	2385	2395	2412	2416	2430
	2440	2450	2454	2491	2498	2501	2507	2516	2521	2530	2533	2546	2548	2557	2612
	2614	2623	2634	2650	2661	2682	2688	2697	2713	2735	2740	2742	2778	2790	2848
	2938	2941	2952	2954	2981	2984	3036	3038	3064	3078	3183	3220	3277	3314	3317
	3321	3385	3432	3434	3490	3505	3548	3567	3569	3571	3701	3703	3838	3944	3949
	3856	3862	3867	3918											
BITB	2958														
BLE	3585														
BLOS	1240	1253	1274	1291	1299	1307	1316	2215	2248	2835	3811	3819	3827	3908	
BLT	2671	3011													
BMI	1159	1429	1441	1478	1520	1526	1561	1744	1833	2068	2162	2561	2933	3004	3014
	3021	3162	3443												
BNE	932	970	1005	1029	1043	1045	1064	1093	1102	1180	1205	1223	1229	1263	1293
	1328	1335	1342	1427	1435	1447	1451	1456	1486	1518	1532	1550	1563	1581	1596
	1623	1637	1641	1644	1655	1753	1786	1813	1819	1825	1839	1866	1904	1906	1932
	1950	1956	1960	1998	2003	2009	2049	2053	2056	2074	2079	2083	2087	2091	2099
	2159	2198	2231	2245	2260	2269	2285	2301	2304	2318	2327	2331	2335	2373	2386
	2417	2441	2492	2508	2522	2531	2534	2549	2558	2579	2589	2611	2613	2651	2662
	2683	2689	2698	2714	2734	2736	2741	2743	2751	2773	2795	2882	2884	2942	2953
	3007	3037	3065	3221	3229	3236	3249	3269	3278	3308	3315	3386	3394	3396	3404
	3415	3435	3506	3517	3549	3560	3597	3607	3674	3699	3702	3704	3774	3796	3803
BPL	963	980	1000	1010	1089	1100	1354	1363	1466	1472	1536	1567	1606	1616	1618
	1627	1629	1663	1697	1699	1710	1723	1725	1738	1790	1792	1798	1800	1823	1870
	1872	1879	1881	1888	1890	1936	1938	1954	1969	1971	2001	2034	2036	2043	2045
	2104	2106	2121	2127	2131	2133	2141	2176	2222	2224	2235	2237	2275	2277	2289

	2291	2325	2367	2369	2381	2382	2407	2409	2425	2427	2582	2592	2594	2627	2754
	2756	2841	2843	2847	2858	2873	2988	2990	3017	3034	3049	3051	3054	3110	3510
SR	3937	3842	3848	3855	3860	3866	3992	3894	3912	3914	3916	3929	3931		
	1011	1020	1026	1097	1123	1152	1191	1246	1356	1431	1437	1449	1453	1458	1464
	1482	1522	1534	1554	1565	1569	1577	1584	1589	1612	1633	1648	1650	1669	1701
	1705	1727	1748	1794	1806	1814	1831	1835	1874	1883	1894	1907	1907	1940	1958
	1962	1973	2007	2039	2057	2077	2081	2085	2089	2226	2241	2249	2255	2305	2312
	2329	2333	2371	2375	2384	2388	2394	2411	2415	2420	2429	2433	2439	2446	2460
	2510	2515	2524	2529	2551	2553	2595	2620	2621	2691	2700	2738	2757	2797	2799
	2859	2879	2885	2901	2908	2957	2962	3019	3022	3041	3052	3080	3090	3097	3102
	3116	3127	3132	3137	3399	3402	3408	3411	3423	3424	3437	3445	3455	3460	3463
CLC	3519	3520	3558	3565	3575	3578	3581	3592	3601	3615	3706	3895	3936	3950	
	1128	1134	1141	1146	1243		1321	1358	1383	1840	1847	2169	2206	2482	2866
CLR	3139	3157	3533	3539	3828										
	933	1021	1059	1118	1119	1160	1178	1184	1194	1195	1197	1198	1199	1200	1201
	1202	1203	1350	1367	1409	1417	1418	1419	1423	1438	1483	1484	1513	1523	1542
	1543	1558	1570	1575	1582	1597	1621	1686	1689	1690	1715	1717	1718	1750	1773
	1776	1777	1778	1782	1809	1815	1816	1909	1924	1925	1928	1929	1946	1964	1965
	1977	1991	1992	2023	2024	2051	2065	2095	2096	2117	2118	2151	2152	2200	2217
	2218	2229	2270	2271	2283	2319	2350	2361	2376	2421	2456	2459	2479	2490	2495
	2486	2539	2540	2543	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572
	2573	2574	2575	2576	2577	2606	2633	2648	2704	2705	2707	2717	2718	2719	2720
	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2768	2777	2788
	2789	2837	2876	2906	2931	2970	3098	3111	3112	3117	3133	3199	3234	3244	3288
	3400	3401	3409	3439	3461	3469	3507	3526	3562	3564	3533	3608	3617	3668	3693
CLRB	3717	3780	3880	3923	3924	3952	3953	3954							
CMP	1025	1027	1047	1103	1555	1658	3128	3456							
	1044	1086	1211	1239	1252	1273	1290	1298	1306	1315	1317	1459	1485	1595	1598
	1636	1640	1654	1702	1728	1785	1812	1818	1895	1931	1949	2052	2055	2098	2214
	2244	2247	2300	2389	2434	2586	2588	2598	2610	2630	2670	2748	2750	2760	2772
	2794	2834	2881	2883	3010	3015	3020	3085	3100	3134	3223	3225	3305	3307	3391
	3393	3414	3419	3421	3516	3559	3584	3588	3673	3698	3795	3802	3810	3818	3826
CMPB	3907														
	1092	1101	1222	1228	1262	1282	1327	1334	1341	1551	1580	1586	1645	2674	2677
	3094	3124	3129	3452	3457										
COM	3609														
DEC	1124	1162	1357	1426	1444	1445	1517	1529	1530	1547	1549	1622	1639	1643	1749
	1855	1905	1997	2071	2072	2158	2230	2268	2284	2303	2956	2964	3088	3122	3228
	3232	3235	3248	3264	3268	3416	3451	3596	3620	3623	3626	3629	3651	3654	3657
	3660	3695	3770	3773											
DECB	2960														
ENT	871														
HAT	915	964	991	2896	2983	3040	3772	3778	3879	3920					
INC	983	1063	1122	1190	1272	1355	1439	1442	1443	1479	1524	1527	1528	1553	1559
	1585	1594	1647	1704	1745	2066	2069	2070	2447	2585	2609	2747	2771	2774	2800
	3087	3099	3121	3126	3136	3166	3444	3454	3462	3483	3492	3550	3591	3744	
INCB	1038	1042	3096	3131	3459										
JMP	1207	1213	1368	1385	1388	1489	1672	1756	1843	1846	1851	2179	2449	2452	2532
	2622	2625	2628	2632	2636	2776	2780	2812	2934	3494	3495	3638	3797	3804	3870
JSR	967	971	977	978	1048	1055	1079	1095	1104	1117	1125	1127	1133	1139	1144
	1155	1175	1176	1196	1215	1218	1219	1221	1225	1227	1232	1234	1235	1248	1250
	1251	1259	1261	1265	1267	1268	1279	1281	1286	1288	1289	1294	1296	1297	1302
	1304	1305	1311	1313	1314	1324	1326	1331	1333	1338	1340	1364	1371	1378	1410
	1413	1416	1454	1509	1512	1514	1556	1611	1624	1656	1668	1682	1685	1687	1688
	1694	1720	1769	1772	1774	1775	1787	1795	1802	1805	1820	1827	1830	1867	1976
	1885	1933	1951	1966	1994	2031	2040	2063	2101	2116	2150	2155	2192	2195	2201



	2219	2232	2261	2262	2263	2272	2286	2322	2363	2377	2404	2422	2475	2478	2493
	2496	2497	2503	2504	2505	2509	2514	2518	2519	2523	2528	2535	2538	2541	2542
	2544	2550	2580	2602	2605	2616	2617	2618	2619	2644	2647	2649	2652	2664	2666
	2685	2686	2690	2695	2699	2706	2708	2737	2744	2764	2767	2801	2804	2808	2829
	2844	2853	2855	2889	2892	2894	2915	2943	2946	2966	2969	2972	2975	2977	2980
	3005	3043	3046	3081	3279	3281	3284	3285	3288	3289	3292	3293	3296	3297	3300
	3301	3304	3310	3313	3319	3323	3326	3327	3330	3331	3334	3335	3338	3339	3342
	3344	3347	3348	3351	3405	3465	3468	3471	3474	3476	3479	3480	3485	3486	3489
	3531	3633	3637	3776	3792	3794	3799	3801	3806	3808	3809	3814	3816	3817	3822
	3824	3825	3832	3833	3877	3903	3905	3906							
MOV	917	935	939	940	941	942	943	944	945	946	950	951	952	953	954
	955	956	957	973	976	1001	1003	1006	1023	1024	1050	1056	1057	1058	1077
	1078	1084	1085	1098	1106	1126	1174	1181	1182	1195	1186	1187	1188	1189	1193
	1209	1217	1241	1271	1275	1276	1292	1300	1308	1323	1351	1366	1373	1377	1381
	1412	1414	1415	1420	1422	1425	1461	1462	1475	1480	1492	1508	1511	1516	1538
	1540	1541	1544	1545	1546	1548	1571	1583	1590	1591	1609	1613	1619	1620	1634
	1635	1649	1651	1652	1666	1681	1684	1691	1692	1693	1713	1716	1719	1730	1731
	1741	1746	1751	1768	1771	1779	1780	1781	1783	1804	1810	1829	1836	1844	1845
	1861	1862	1863	1864	1875	1884	1897	1898	1901	1902	1908	1913	1926	1927	1941
	1942	1943	1945	1947	1963	1974	1982	1990	1993	1996	2012	2022	2025	2026	2027
	2028	2029	2030	2050	2058	2061	2093	2100	2108	2115	2124	2128	2144	2153	2154
	2157	2166	2168	2174	2191	2194	2196	2199	2204	2209	2210	2211	2212	2216	2227
	2228	2242	2243	2250	2251	2253	2264	2265	2266	2279	2281	2282	2295	2297	2298
	2299	2306	2307	2309	2320	2321	2337	2343	2344	2348	2349	2351	2352	2353	2354
	2355	2356	2357	2358	2359	2360	2362	2391	2392	2398	2400	2401	2402	2403	2436
	2437	2444	2457	2458	2474	2477	2481	2484	2487	2488	2489	2490	2490	2500	2526
	2554	2555	2556	2559	2600	2604	2607	2643	2646	2655	2656	2657	2658	2659	2660
	2663	2667	2668	2672	2680	2681	2684	2693	2702	2710	2711	2712	2715	2762	2756
	2769	2792	2803	2805	2826	2827	2828	2830	2832	2833	2836	2838	2839	2851	2864
	2865	2871	2877	2880	2886	2887	2891	2900	2905	2907	2916	2917	2918	2919	2921
	2935	2945	2947	2949	2950	2963	2968	2974	2979	2993	2994	3012	3024	3025	3026
	3027	3028	3029	3045	3047	3055	3056	3067	3068	3082	3083	3092	3115	3118	3119
	3138	3154	3160	3169	3170	3171	3172	3173	3174	3175	3181	3182	3189	3196	3217
	3218	3219	3222	3227	3231	3239	3240	3241	3242	3243	3262	3263	3265	3267	3293
	3287	3291	3295	3299	3303	3312	3325	3329	3333	3337	3341	3346	3350	3382	3383
	3384	3387	3389	3390	3397	3398	3406	3407	3410	3412	3413	3428	3429	3440	3447
	3448	3467	3473	3478	3482	3484	3488	3511	3512	3513	3514	3525	3530	3535	3537
	3541	3542	3551	3552	3553	3554	3555	3563	3566	3574	3577	3580	3583	3586	3587
	3594	3599	3600	3602	3603	3605	3610	3612	3613	3614	3616	3619	3622	3625	3628
	3631	3636	3649	3650	3653	3656	3659	3662	3669	3680	3685	3686	3691	3694	3697
	3705	3707	3714	3715	3716	3812	3820	3831	3834	3840	3851	3852	3858	3869	3881
	3882	3883	3886	3887	3890	3921	3925	3926	3934	3944	3945	3947	3948	3949	3951
	3955														
MOV B	975	1008	1019	1022	1046	1090	1161	1352	1421	1470	1578	1579	1604	1661	1708
	1736	1944	1999	2119	2125	2129	2139	2160	2590	2752	2971	2976	2986	3069	3201
	3470	3475	3846	3864	3909	3910									
NEG	2920														
NEGB	3721	3722	3724	3725	3727	3728	3730	3731	3733	3734	3736	3737	3739	3740	3742
	3743	3746	3747	3749	3750	3752	3753	3755	3756	3758	3759	3761	3762	3764	3765
	3767	3768													
NOP	2494	2495	2536	2537	2653	2654	2809	2810	2811						
RESET	1173	2165	2807	3791	3876										
RCL	1030	1032	1034	1061	1062	1129	1130	1131	1142	1147	1148	1149	1150	1244	1245
	1255	1256	1322	1359	1360	1384	1841	3140	3141	3142	3143	3144	3145	3146	3147
	3246	3247	3829	3830											
ROLB	1031	1033	1035	3720	3723	3726	3729	3732	3735	3738	3741	3745	3748	3751	3754



ROR	3757	3760	3763	3766	3769										
RTI	1136	1137	1374	1375	2170	2171	2207	2483	2867	2868	3155	3156	3197	3198	3534
RTS	934	936	984	1493	2995	3058									
	998	1007	1015	1018	1051	1080	1107	1156	1164	2925	3057	3066	3070	3149	3164
	3176	3190	3202	3230	3237	3270	3316	3352	3425	3543	3664	3676	3708	3781	
SEC	3718														
SUB	974	1238	2205	2578	2733	3013	3441	3442	3524	3529	3532	3573	3576	3579	3582
SWAB	1135	1140	1145	2172	2869	3158	3188	3684	3690	3885					
TRAP	869														
TST	931	962	979	1158	1183	1236	1269	1348	1353	1362	1432	1440	1446	1450	1465
	1477	1525	1531	1535	1562	1566	1573	1592	1617	1628	1638	1642	1698	1724	1743
	1752	1784	1791	1799	1807	1811	1817	1832	1838	1871	1880	1889	1903	1930	1937
	1948	1970	1975	2035	2044	2073	2105	2132	2175	2213	2223	2236	2246	2267	2276
	2290	2302	2311	2345	2368	2381	2408	2426	2560	2583	2593	2596	2626	2745	2755
	2758	2842	2872	2874	2937	2936	2989	2991	3006	3018	3023	3033	3050	3053	3161
	3395	3403	3426	3430	3449	3515	3527	3556	3606	3672	3675	3841	3859	3891	3913
	3915	3930													
TSTB	999	1004	1009	1036	1039	1088	1099	1120	1428	1471	1519	1560	1605	1615	1626
	1662	1696	1709	1722	1737	1789	1797	1822	1869	1878	1887	1935	1953	1968	2000
	2033	2042	2067	2103	2120	2126	2130	2140	2161	2221	2234	2274	2288	2324	2366
	2379	2406	2424	2581	2591	2753	2840	2846	2857	2987	3003	3048	3109	3509	3836
	3847	3854	3865	3893	3911	3928									
WAIT	2511	2525	2552	2692	2701	2739	3032								
.ABS	776														
.ASCII	4044	4171	4178	4182	4260	4261	4268	4271	4276	4281	4287	4294	4300	4305	4312
	4319	4330	4340	4349	4357	4365									
.ASCIZ	989	990	1110	1111	3960	3963	3966	3969	3971	3973	3975	3978	3982	3990	3993
	3999	4002	4007	4011	4015	4017	4021	4024	4027	4030	4033	4036	4038	4041	4047
	4052	4055	4059	4062	4066	4070	4079	4084	4090	4096	4103	4107	4114	4119	4123
	4133	4137	4141	4146	4150	4155	4159	4162	4164	4186	4194	4201	4371		
.BLKW	1052	1109													
.BYTE	3206														
.ENABL	1169														
.END	4380														
.ENDC	979														
.EVEN	992	3209	3358	3959	4177	4203	4259								
.IF	978														
.LIST	774	915	1169												
.MACR	867	898	899	900	901	902	903	904	905	906	907	909	910	911	912
	913														
.MACRO	908	3710													
.NLIST	775	915	1169												
.PAGE	1169	2180	2815	2910											
.REPT	1	915	2562	2717	3719	3745									
.SBTTL	1401	1496	1673	1758	2180	2461	2637	2815	2910						
.TITLE	777	1169													
.WORD	1557	1625	1695	1721	1788	1796	1821	1868	1877	1886	1934	1952	1967	1995	2032
	2041	2064	2102	2156	2220	2233	2273	2287	2323	2364	2378	2405	2423	2506	2520
	2545	2687	2696	2709	2845	2856									

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

G09

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:19 PAGE 113  
DZRPB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

\*.DZRPB.SEG/SOL/CRF/PAGNUM-DZRPB.P11,DZRPB.SRC  
RUN-TIME: 15 29 6 SECONDS  
RUN-TIME RATIO: 258/52=4.9  
CORE USED: 15K (29 PAGES)

