

RP11-C/RP02

RELIABILITY
MD-11-DZRPF-B

EP-DZRPF-B-DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

This microfiche card contains a grid of frames. The frames are arranged in approximately 10 rows and 10 columns. Each frame contains a small, high-contrast image or data set, likely representing a specific data point or a small table. The overall appearance is that of a dense data matrix. The frames are separated by thin white lines, and the entire grid is set against a dark background.



1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043

1.0 ABSTRACT

THIS PROGRAM TESTS BOTH THE ADDRESSING CAPABILITY AND THE DATA RELIABILITY OF THE OF THE P11 AND THE RPO2. 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
RP11 DISK PACK CONTROLLER WITH UP TO EIGHT RPO2 DRIVES
ASR33 OR EQUIVELANT.

2.2 STORAGE

8K OF STORAGE IS REQUIRED TO RUN THIS PROGRAM.

2.3 PRELIMINARY PROGRAMS

DZRPE RP11 DISKLESS DIAGNOSTIC

3.0 LOADING PRODEDURE

USE STANDARD PROCEDURE FOR ABS TAPES

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

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

E01

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 4
DZRPFB.P11

144

4.2 STARTING ADDRESS

f

145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

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

5.0 OPERATING PROCEDURES

5.1 OPERATIONAL SWITCH SETTINGS

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

5.1.1 SWITCH SETTINGS ARE:

- SW<15>=1....HALT ON ERROR
- SW<14>=1....LOOP ON ERROR
- SW<13>=1....INHIBIT PRINTOUT
- SW<12>=1....INHIBIT BACKGROUND TEST
- SW<11>=1....RING BELL ON ERROR
- SW<10>=1....LOOP ON TEST
- SW<09>=1....INHIBIT DATA COMPARISON
- SW<08>=1....ENTER CONVERSATION MODE

GO1

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 6
DZRPFB.P11

201
202

SW<05>

USED TO CONTROL HOW MANY COMPARE ERRORS
WILL BE TYPED OUT AS A RESULT OF A READ

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
249
250
251
252

OPERATION IN THE DATA TEST.

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

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

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

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

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

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

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

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

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

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

5.2.2 HLT

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

5.2.3 BACKGROUND TEST

THIS TEST IS ENTERED BY THE PROGRAM WHILE WAITTING FOR AN INTERRUPT. IT DOES A SERIES OF NEGATE BYTE INSTRUCTIONS TO PROVIDE WORSE CASE NPR TIMING 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.

294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349

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 AND RPCS 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=
STATUS ERROR
RPDS=
RPER=
RPCS=
CYLINDER=

HEAD=

SECTOR= | PC OF FAILURE

CONTENTS OF RPDS
CONTENTS OF RPER
CONTENTS OF RPCS
CYLINDER ADDRESS OF
THE ERROR
HEAD ADDRESS OF THE
ERROR
SECTOR ADDRESS OF
THE ERROR |
| 2. | PC=
COMPARE ERROR
EXPECTED=
RECEIVED=
CYLINDER=

HEAD=

SECTOR=

WORD COUNT INTO SECTOR=

READ NO. | PC OF FAILURE

DATA EXPECTED
DATA RECEIVED
CYLINDER ADDRESS OF
THE ERROR
HEAD ADDRESS OF THE
ERROR
SECTOR ADDRESS OF
THE ERROR
DISTANCE INTO
SECTOR. THE COUNT
STARTS AT ONE.
INDICATES WHICH READ
ATTEMPT IS IN
PROGRESS. |
| 3. | TOTAL REREADS ON ERROR= | TOTAL READS BEFORE |

K01

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 10
DZRPFB.P11

350
351

RECOVERY. TOTAL OF
24 INDICATES ERROR

WAS UNRECOVERABLE

352
353
354
355
356
357
358
359
360
361
362
363
364
365
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

7.0 RESTRICTIONS

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

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

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

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500.

9.0 PROGRAM DISCRIPTION

9.1 ADDRESS TEST 0

IN THIS TEST THE PROGRAM SEEKS FROM 0 TO N AND BACK
TO 0. N STARTS AT ZERO THEN INCREMENTS TO 1 AND UP
THRU 312. DONE IS TIMED OUT, SEEK UNDERWAY IS
TESTED, UNIT READY IS TIMED OUT. ATTENTION
INTERRUPT AND THE INTERRUPT FLAG ARE TESTED, AND THE
CONTENTS OF THE SELECTED CYLINDER REGISTER ARE
CHECKED.

9.2 ADDRESS TEST 1

MO1

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

408
409

WRITE 5000(OCTAL) WORDS IN THE TEN SECTORS ON EACH

410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465

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

9.3 ADDRESS TEST 2

WRITE THE FIRST WORD OF EACH CYLINDER WITH THE CYLINDER ADDRESS. THEN SEEK FROM 312 TO 0. THEN SEEK TO 311 AND BACK TO 1 UNTIL THE ADDRESSES CROSS AND GO BACK TO ZERO. AFTER EACH SEEK VERIFY THE POSITION OF THE HEAD BY READING THE FIRST WORD OF THE CYLINDER AND COMPARING IT WITH THE SELECTED CYLINDER.

9.4 WRITECHECK TEST

THIS TEST VERIFIES THE WRITE CHECK LOGIC AND THE ABILITY OF THE HARDWARE TO FILL THE REMAINDER OF A SECTOR WITH ZEROS WHEN A PARTIAL SECTOR IS WRITTEN. IN THE WRITE CHECK PORTION A FLOATING ONE AND A FLOATING ZERO PATTERN ARE USED TO TEST THE WRITE CHECK COMPARE LOGIC. THE PATTERN IS WRITTEN AND WRITE CHECKED EXPECTING NO ERRORS. THE BUFFER IS THEN CLEARED AND THE DATA IS WRITE CHECKED AGAIN. AN ERROR IS EXPECTED. AFTER WRITE CHECK IS TESTED, A SECTOR IS WRITTEN WITH ALL ONES AND THEN A TWO WORD WRITE IS PERFORMED. THE ENTIRE SECTOR IS READ AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND THE REMAINDER SHOULD BE ZERO.

9.5 MEMORY TEST

THIS TEST 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)

B02

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 14
DZRPFB.P11

456
467

IS WRITTEN WITH ITS ADDRESS. THIS DATA IS WRITTEN
ON THE DISK. THE MEMORY IS CLEARED AND THE DATA IS

468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523

READ BACK AND VERIFIED. 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
DRIVE. THE SEQUENCE IS WRITE THE PACK, WRITE CHECK,
AND READ IT. THIS SEQUENCE CONTINUES FOR THE 15
PATTERNS DEFINED BELOW. IF A DATA ERROR IS
ENCOUNTERED DURING A READ OPERATION, THE OPERATION
IS REPEATED 20 TIMES OR UNTIL THE ERROR GOES AWAY.
AFTER THE TENTH TIME THE HEADS ARE HOMED AND
REPOSITIONED. WITH EACH READ ERROR THE READ RETRY
NUMBER IS TYPED OUT ALONG WITH THE ERROR
INFORMATION. THIS WAY IT CAN BE DETERMINED IF AN
ERROR IS RECOVERABLE OR NON-RECOVERABLE. IF A READ
STATUS ERROR OCCURS AND IT IS A SOFT ERROR (PARITY
ERROR), THE DATA IS COMPARED TO PROVIDE ADDITIONAL
INFORMATION.

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 400(OCTAL) WORDS IS
WRITTEN ON RANDOM SECTORS. IT IS THEN WRITE CHECKED
AND READ. THIS IS REPEATED 5000 TIMES. THE READ
RECOVERY TECHNIQUE IS THE SAME AS DESCRIBED UNDER
DATA TEST(9.4).

9.8 POWER FAIL TEST

D02

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

524
525

TESTS THE ABILITY OF THE RP11 TO SENSE POWER FAILURE
AND TO HOME THE HEADS. AS SOON AS THE OPERATOR IS

526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581

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

9.9 CONVERSATION MODE

CONVERSATION MODE MAY BE ENTERED BY SETTING SW<08> TO A 1. IF SELECTED A NUMBER OF QUESTIONS WILL BE ASKED TO DETERMINE TEST PARAMETERS. ALL NUMBER RESPONSES SHOULD BE IN OCTAL FOLLOWED BY A CARRIAGE RETURN.

THE CONVERSATION IS AS FOLLOWS:

DATA TEST ONLY?(Y OR N)

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

MULTI DRIVE MODE?(Y OR N)

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

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

NUMBER OF DRIVES 1 TO 10 (OCTAL)?

RESPOND WITH THE NUMBER OF DRIVES ON THE SYSTEM.

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

WHICH DRIVE?

RESPOND WITH UNIT NUMBER OF DRIVE TO TEST

OPTIONAL WORD COUNT?(Y OR N)

F02

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 18
DZRPFB.P11

582
583

PROVIDES THE OPPORTUNITY TO SPECIFY YOUR OWN WORD
COUNT WHICH MAY NOT EXCEED THE STANDARD WORD COUNT.

584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639

THE STANDARD WORD COUNT IS TYPED OUT AT THE START OF THE PROGRAM. IF RESPONSE IS NO - THE NEXT QUESTION IS SKIPPED.

LENGTH (1 TO STANDARD WORD COUNT)?

SPECIFY WORD COUNT IN OCTAL.

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

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

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

STARTING CYLINDER
STARTING HEAD
STARTING TRACK

RESPOND WITH THE DESIRED ADDRESS IN OCTAL

OPTIONAL DATA PATTERN NO.?

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

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

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

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

%

.LIST ME
.NLIST MC,MD,CND
.ABS
.TITLE FRONT END

```

640
641           ;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS
642
643           ;GENERAL REGISTER ASSIGNMENTS
644           R0=%0
645           R1=%1
646           R2=%2
647           R3=%3
648           R4=%4
649           R5=%5
650           SP=%6
651           PC=%7
652
653           ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
654           C=1           ;C BIT
655           V=2           ;V BIT
656           Z=4           ;Z BIT
657           N=10          ;N BIT
658           T=20          ;T BIT
659           PRI7=340      ;PRIORITY LEVEL 7
660           PRI6=300      ;PRIORITY LEVEL 6
661           PRI5=240      ;PRIORITY LEVEL 5
662           PRI4=200      ;PRIORITY LEVEL 4
663           PRI3=140      ;PRIORITY LEVEL 3
664           PRI2=100      ;PRIORITY LEVEL 2
665           PRI1=40       ;PRIORITY LEVEL 1
666
667           ;VECTOR ADDRESSES
668           ERRVEC=4       ;ERROR VECTOR
669           RESVEC=10      ;RESERVED INST VECTOR
670           TBITVEC=14     ;T BIT VECTOR
671           IOTVEC=20      ;IOT TRAP VECTOR
672           PFVEC=24       ;POWER FAIL VECTOR
673           EMTVEC=30      ;EMT VECTOR
674           TRAPVEC=34    ;TRAP VECTOR
675
676           ;REGISTER ADDRESSES
677           PSW=177776     ;PROCESSOR STATUS REGISTER
678           TKS=177560     ;KEYBOARD CSR
679           TKB=177562     ;ADDR OF KEYBOARD BUFFER
680           TPS=177564     ;TELEPRINTER CSR
681           TPB=177566     ;TELEPRINTER BUFFER
682           SWR=177570     ;CONSOLE SWITCH REGISTER
683           DISPLAY=177570 ;CONSOLE DISPLAY REGISTER
684
685           ;INITIAL STACK POINTER
686           STKPTR=500     ;PROGRAM STACK POINTER
687
688           ;BIT ASSIGNMENTS
689           B15=100000
690           B14=40000
691           B13=20000
692           B12=10000
693           B11=4000
694           B10=2000
695           B9=1000

```

FRONT END
DZRPFB.P11

MACY11 27(732) 16-SEP-76 16:12 PAGE 21

696	000400	B8=400
697	000200	B7=200
698	000100	B6=100
699	000040	B5=40
700	000020	B4=20
701	000010	B3=10
702	000004	B2=4
703	000002	B1=2
704	000001	B0=1

;MEMORY MANAGEMENT REGISTER ASSIGNMENTS

708	177572	SRO=177572
709	172340	KIPAR0=172340
710	172342	KIPAR1=172342
711	172344	KIPAR2=172344
712	172356	KIPAR7=172356
713	172300	KIPDR0=172300
714	172302	KIPDR1=172302
715	172304	KIPDR2=172304
716	172316	KIPDR7=172316
717	000006	RW=6
718	000000	UP=00

;INSTRUCTION EQUATES

721		HLT=TRAP
722	104400	
723		SCOPE=EMT
724	104000	

;HLT IS A TRAP TO THE ERROR ROUTINE

;SCOPE IS AN EMT TRAP

;INDEX OF MACROS

728	.SCOPE
729	.SAVE
730	.REST
731	.ERROR
732	.PRINT
733	.DUMP
734	.RAND
735	.READ
736	.PACK

;INDEX OF CALLS

738	SCOPE
739	SAVE
740	REST
741	HLT
742	PRINT
743	DUMP
744	DUMPF
745	SDUMP
746	SDUMPF
747	RAND
748	READ
749	PACK

751

752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807

000046
000046 000046
000052 012060
000052 000052
000052 040000
000200 000200
000200 012707 002336
001000 001000
001002 000000
001004 032737 040000 177570
001012 001403
001014 005767 000220
001020 100403
001022 005067 000212
001026 000002
001030 016716 177746
001034 000002
001036 012667 000020
001042 010546
001044 010446
001046 010346
001050 010246
001052 010146
001054 010046
001056 016707 000000
001062 000000
001064 012667 000020

```

      . =46
      MEXIT
      . =52
      40000
      .LIST ME
      . =200
      MOV #START,PC ;GO TO START OF TEST
      . =1000
ICNT: 0 ;CONTAINS PASS COUNT
LAD: 0 ;PROGRAM TRACE
;SCOPE (EMT) SERVICE ROUTINE
;THIS ROUTINE WILL LOOP IF AN ERROR OCCURED AND
;LOOP ON ERROR SWITCH IS SET (BIT 14). IF LOOPING IS INDICATED
;THE CONTENTS OF "LAD" EQUAL THE LOOP ADDRESS. IN ORDER
;TO LOOP ON ERROR, BIT 14 OF THE SWITCH REGISTER MUST BE SET AND
;LOCATION "ERRFLG" MUST BE NEGATIVE INDICATING AN ERROR. ONCE THE
;LOOP IS INITIATED IT WILL CONTINUE UNTIL SWITCH 14 IS CLEARED.
SCOPE: BIT #B14,2#SWR ;LOOP ON ERROR?
      BEQ 2$ ;BRANCH IF NO
      TST ERRFLG ;IS THERE AN ERROR?
      BMI 1$ ;BRANCH IF YES
2$: CLR ERRFLG ;RESET ERROR CONDITION
      RTI ;EXIT
1$: MOV LAD,(SP) ;MODIFY RETURN ADDRESS
      RTI ;EXIT
;ROUTINE TO SAVE REGISTERS ON THE STACK.
;CALLED BY SAVE MACRO
SAVES: MOV (SP)+,1$ ;SAVE RETURN PC
      MOV R5,-(SP)
      MOV R4,-(SP)
      MOV R3,-(SP)
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV R0,-(SP)
      MOV 1$,PC ;RETURN
1$: 0 ;CONTAINS RETURN ADDRESS
;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
;CALLED BY REST MACRO
RESTS: MOV (SP)+,1$ ;SAVE RETURN PC
```

```

808 001070 012600      MOV      (SP)+,R0
809 001072 012601      MOV      (SP)+,R1
810 001074 012602      MOV      (SP)+,R2
811 001076 012603      MOV      (SP)+,R3
812 001100 012604      MOV      (SP)+,R4
813 001102 012605      MOV      (SP)+,R5
814 001104 016707 000000  MOV      1$,PC
815 001110 000000      ;RETURN
816      ;CONTAINS RETURN ADDR
817      ;ERROR SERVICE ROUTINE CALLED BY HLT
818      ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
819      ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
819 001112 005737 177570  ERROR:  TST      @#SWR      ;HALT ON ERROR?
820 001116 100001      BPL      3$      ;BRANCH IF NO
821 001120 000000      HALT
822 001122 032737 004000 177570  3$:  BIT      #B11,@#SWR      ;RING THE BELL?
823 001130 001403      BEQ      1$      ;BRANCH IF NO
824 001132 004567 000144      JSR      R5,PRNTFS      ;FORCE PRINT THE MESSAGE
825 001136 001250      BELL
826 001140 032737 020000 177570  1$:  BIT      #B13,@#SWR      ;SKIP TYPEOUT?
827 001146 001022      BNE      2$      ;BRANCH IF YES
828 001150 004567 000110      JSR      R5,PRINTS      ;PRINT MESSAGE
829 001154 001252      ERRPC
830 001156 011667 000062      MOV      (6),HLTADS      ;GET ERROR PC+2
831 001162 162767 000002 000054      SUB      #2,HLTADS      ;MODIFY
832 001170 117767 000050 000044      MOV      @HLTADS,HLTCTS      ;SAVE HLT ARGUMENT
833 001176 016767 000042 000356      MOV      HLTADS,TTY
834 001204 004767 000134      JSR      PC,PRINTR      ;TYPE LOCATION WITH LEADING ZEROS
835 001210 004767 013146      JSR      PC,MSG      ;GO TO USER ERROR ROUTINE
836 001214 005737 177570  2$:  TST      @#SWR      ;HALT ON ERROR?
837 001220 100001      BPL      4$      ;BRANCH IF NO
838 001222 000000      HALT
839 001224 052767 100000 000006  4$:  BIS      #B15,ERRFLG      ;SET ERROR FLAG
840 001232 005267 000010      INC      ERRORS      ;UPDATE ERROR COUNTER
841 001236 000002      RTI
842 001240 000000      ERRFLG: 0
843 001242 000000      HLTCTS: 0
844 001244 000000      HLTADS: 0      ;PC OF ERROR
845 001246 000000      ERRORS: 0      ;ERROR COUNT
846 001250 000007      BELL: .ASCIZ <7>
847 001252 005015 005015 041520  ERRPC: .ASCIZ <15><12><15><12>'PC= '
848 001260 020075 000      .EVEN
849 001264
850      ;THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
851      ;THE MESSAGE MUST TERMINATE IN 0
852 001264 032737 020000 177570  PRINTS: BIT      #B13,@#SWR      ;INHIBIT TYPEOUTS?
853 001272 001403      BEQ      PRNTFS      ;BRANCH IF NO
854 001274 062705 000002      ADD      #2,R5      ;UPDATE RETURN ADDR
855 001300 000205      RTS      R5
856 001302 105737 177564      PRNTFS: TSTB     @#TPS      ;WAIT FOR PRINTER TO FINISH
857 001306 100375      BPL      -4
858 001310 010546      MOV      R5,-(SP)
859 001312 062716 000002      ADD      #2,(SP)      ;ADJUST RETURN PC
860 001316 011505      MOV      (R5),R5      ;GET MESSAGE ADDR
861 001320 105715  1$:  TSTB     (R5)      ;CHECK FOR TERMINATOR
862 001322 001002      BNE      2$
863 001324 012605      MOV      (SP)+,R5      ;GET RETURN ADDR

```



```

864 001326 000205          RTS      R5          ;RETURN
865 001330 112537 177566 2$:      MOVB   (R5)+,2#TPB ;PRINT CHARACTER
866 001334 105737 177564          TSTB   2#TPS      ;WAIT TILL DONE
867 001340 100375          BPL     -4
868 001342 000766          BR      1$
869          ;THIS ROUTINE TYPES A LOCATION IN OCTAL
870 001344 032737 020000 177570 PRINTR: BIT   #B13,2#SWR ;INHIBIT TYPEOUT?
871 001352 001406          BEQ     PRINTA    ;BRANCH IF NO
872 001354 000207          RTS     PC
873 001356 032737 020000 177570 PRINTS: BIT   #B13,2#SWR ;INHIBIT TYPEOUT?
874 001364 001405          BEQ     PRINTB    ;BRANCH IF NO
875 001366 000207          RTS     PC
876 001370 112767 000001 000140 PRINTA: MOVB  #1,.PR ;SET ZERO FILL SWITCH
877 001376 000402          BR      .+6      ;SKIP
878 001400 005067 000132          PRINTB: CLR  .PR ;SUPPRESS LEADING ZEROS
879 001404 112767 177772 000125          MOVB  #-6,.PR+1 ;SET COUNT
880 001412 010446          .PTIT: MOV   R4,-(SP) ;SAVE R4
881 001414 012704 001540          MOV   #.PR+2,R4 ;SET POINTER TO FIRST CHARACTER
882 001420 105014          CLRB  (R4)      ;CLEAR FIRST BYTE
883 001422 000413          BR      .PRF    ;ROTATE FIRST BIT
884 001424 105014          .PRL: CLRB  (R4) ;CLEAR BYTE OF CHAR
885 001426 032767 000100 000102          BIT   #100,.PR ;BIT TYPING MODE
886 001434 001006          BNE   .PRF      ;YES SKIP 2 ROTATES
887 001436 006167 000120          ROL   TTY      ;ROTATE BIT INTO C
888 001442 106114          ROLB  (4)      ;PACK IT
889 001444 006167 000112          ROL   TTY
890 001450 106114          ROLB  (4)
891 001452 006167 000104          .PRF: ROL   TTY
892 001456 106114          ROLB  (4)
893 001460 105714          TSTB  (4)      ;IS IT ZERO
894 001462 001402          BEQ   .+6      ;SKIP INC
895 001464 105267 000046          INCB  .PR      ;SET FILL SWITCH
896 001470 105767 000042          TSTB  .PR      ;CHECK FILL SWITCH
897 001474 001402          BEQ   .+6      ;SKIP BITSET
898 001476 152724 000060          BISB  #'0,(4)+ ;MAKE INTO ASCIZ CHAR
899 001502 105267 000031          INCB  .PR+1    ;INC COUNT
900 001506 001346          BNE   .PRL     ;REPEAT
901 001510 022704 001540          CMP   #.PR+2,R4 ;EMPTY BUFFER
902 001514 001002          BNE   .+6      ;SKIP IF NOT
903 001516 112724 000060          MOVB  #'0,(4)+ ;LOAD ONE ZERO
904 001522 105014          CLRB  (4)      ;NULL TERMINATOR
905 001524 004567 177534          JSR   R5,PRINT$ ;PRINT MESSAGE
906 001530 001540          .PR+2
907 001532 012604          MOV   (SP)+,R4 ;RESTORE R4
908 001534 000207          RTS   PC
909 001536 000012          .PR:  .BLKW  12
910 001562 000000          TTY:  0
911 001564          RAND$:
912 001564 004767 177246          JSR   PC,SAVES ;SAVE THE REGISTERS
913 001570 016700 000106          MOV   LONUM,R0 ;SET R0 WITH LOW
914 001574 016701 000100          MOV   HINUM,R1 ;SET R1 WITH HIGH
915 001600 012703 177771          MOV   #-7,R3   ;SET SHIFT COUNT
916 001604 005002          CLR   R2
917 001606 006300          1$:  ASL   R0      ;SHIFT R0 LEFT AND
918 001610 006101          ROL   R1      ;ROTATE CARRY INTO R1 AND
919 001612 006102          ROL   R2      ;ROTATE CARRY INTO R2

```

```

920 001614 005203          INC      R3          ;CHECK FOR DONE
921 001616 001373          BNE     1$
922 001620 066702 000056  ADD     LONUM,R2    ;ADD # TO MAKE X 129
923 001624 005501          ADC     R1          ;PROPOGATE CARRY
924 001626 066701 000046  ADD     HINUM,R1    ;ADD # TO MAKE X 129
925 001632 005502          ADC     R2          ;PROPOGATE CARRY
926 001634 062700 001057  ADD     #1057,R0
927 001640 005501          ADC     R1          ;PROPOGATE CARRY
928 001642 005502          ADC     R2          ;PROPOGATE CARRY
929 001644 062701 047401  ADD     #47401,R1
930 001650 005502          ADC     R2
931 001652 062702 000006  ADD     #6,R2
932 001656 060200          ADD     R2,R0
933 001660 005501          ADC     R1
934 001662 010067 000014  MOV     R0,LONUM
935 001666 010167 000006  MOV     R1,HINUM
936 001672 004767 177166  JSR     PC,RESTS    ;RESTORE THE REGISTERS
937 001676 000207          RTS     PC
938
939 001700 000000          HINUM:  0
940 001702 000000          LONUM:  0
941 001704 010346          READS:  MOV     R3,-(6)    ;SAVE R3
942 001706 012703 002014  1$:     MOV     #INPUTS,R3 ;GET BUFFER ADDR
943 001712 022703 002034  2$:     CMP     #INPUTS+20,R3 ;BUFFER FULL?
944 001716 001412          BEQ     4$          ;YES..TYPE ?
945 001720 105737 177560  TSTB   @#177560    ;WAIT FOR A CHAR
946 001724 100375          BPL     -4
947 001726 113713 177562  MOVB   @#177562,(3) ;GET CHAR
948 001732 142713 000200  BICB   #200,(3)    ;GET RID OF JUNK
949 001736 122713 000177  CMPB   #177,(3)    ;IS IT A RUBOUT?
950 001742 001004          BNE     3$          ;SKIP IF NO
951 001744          4$:
952 001744 004567 177314  JSR     R5,PRINTS  ;PRINT MESSAGE
953 001750 002054          READMS
954 001752 000755          BR     1$
955 001754 013737 177562 177566 3$:     MOV     @#TKB,@#TPB ;CLEAR BUFFER AND START OVER
956 001762 105737 177564  TSTB   @#TPS      ;ECHO THE CHAR
957 001766 100375          BPL     -4          ;WAIT FOR READY
958 001770 122723 000015  CMPB   #15,(3)+    ;CHECK FOR RETURN
959 001774 001346          BNE     2$          ;LOOP IF NOT RETURN
960 001776 105063 177777  CLRB   -1(3)       ;REMOVE THE RETURN
961 002002 004567 177256  JSR     R5,PRINTS  ;PRINT MESSAGE
962 002006 002060          READLS
963 002010 012603          MOV     (6)+,R3    ;RESTORE R3
964 002012 000207          RTS     PC         ;RETURN
965
966 002014 000020          INPUTS: .BLKW 20
967 002054 006477 000012  READMS: .ASCIZ '??'<15><12>
968 002060 000012          READLS: .ASCIZ <12>
969
970          ;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
971          ;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
972
973 002062          PACKS:
974 002062 004767 176750  JSR     PC,SAVE$   ;SAVE THE REGISTERS
975 002066 005067 000242  CLR     NUM$

```

```

976 002072 005000
977 002074 105760 002014 2S: CLR RO
978 002100 001402 BEQ INPUTS(RO)
979 002102 005200 INC 1S
980 002104 000773 BR 2S
981 002106 005300 1S: DEC RO
982 002110 004767 000166 JSR PC,PACS ;GET OCTAL CHAR
983 002114 016767 000212 000212 MOV PK$,NUMS ;PACK FIRST CHAR
984 002122 004767 000154 JSR PC,PACS ;GET OCTAL CHAR
985 002126 000241 CLC
986 002130 006167 000176 ROL PK$
987 002134 006167 000172 ROL PK$
988 002140 006167 000166 ROL PK$
989 002144 056767 000162 000162 BIS PK$,NUMS ;PACK SECOND CHAR
990 002152 004767 000124 JSR PC,PACS ;GET OCTAL CHAR
991 002156 000241 CLC
992 002160 000367 000146 SWAB PK$
993 002164 006067 000142 ROR PK$
994 002170 006067 000136 ROR PK$
995 002174 056767 000132 000132 BIS PK$,NUMS ;PACK THIRD CHAR
996 002202 004767 000074 JSR PC,PACS ;GET OCTAL CHAR
997 002206 000367 000120 SWAB PK$
998 002212 000241 CLC
999 002214 006167 000112 ROL PK$
1000 002220 056767 000106 000106 BIS PK$,NUMS ;PACK FOURTH CHAR
1001 002226 004767 000050 JSR PC,PACS ;GET OCTAL CHAR
1002 002232 000367 000074 SWAB PK$
1003 002236 000241 CLC
1004 002240 006167 000066 ROL PK$
1005 002244 006167 000062 ROL PK$
1006 002250 006167 000056 ROL PK$
1007 002254 006167 000052 ROL PK$
1008 002260 056767 000046 000046 BIS PK$,NUMS ;PACK FIFTH CHAR
1009 002266 000402 BR PKEX1$
1010 002270 062706 000002 PKEX$: ADD #2,SP ;MODIFY STACK
1011 002274 PKEX1$:
1012 002274 004767 176564 JSR PC,RESTS ;RESTORE THE REGISTERS
1013 002300 000207 RTS PC ;EXIT
1014
1015 002302 005700 PAC$: TST RO
1016 002304 100771 BMI PKEX$
1017 002306 005067 000020 CLR PK$
1018 002312 116067 002014 000012 MOVB INPUTS(RO),PK$ ;GET INPUT CHAR
1019 002320 005300 DEC RO
1020 002322 042767 177770 000002 BIC #177770,PK$ ;CLEAR UNWANTED BITS
1021 002330 000207 RTS PC
1022
1023 002332 000000 PK$: 0
1024 002334 000000 NUM$: 0
1025 .TITLE RP11C RELIABILITY TEST
1026
1027 000254 VECTOR=254 ;DISK INTERRUPT TRAP ASSIGNMENT
1028 000256 STATUS=256 ;INTERRUPT PRIORITY ASSIGNMENT
1029
1030
1031 002336 000005 START: RESET ;CLEAR THE WORLD

```

Line No.	Address	Op Code	Operand 1	Operand 2	Comment
1032	002340	012706	000500		MOV #STKPTR, SP ; SETUP STACK
1033	002344	004767	011352		JSR PC, INIT ; INITIALIZE VECTORS
1034	002350	004567	013102		JSR RS, EXTMEN ; SET UP DATA BUFFERS
1035	002354	005067	176420		CLR ICNT ; CLEAR THE PASS COUNTER
1036	002360	005067	015100		CLR FLAG ; CLEAR PROGRAM FLAG
1037	002364	005067	015130		CLR DSKNOR ; CLEAR UNIT FLAG
1038	002370	005067	015102		CLR CYLINDER ; CLEAR THE CYLINDER ADDRESS
1039	002374	005067	015100		CLR DMA ; CLEAR DAR REGISTERS
1040	002400	005067	015100		CLR PATNU ; CLEAR PATTERN COUNT
1041	002404	032737	000400	177570	BIT #88, 2#SWR ; USE CONVERSATION MODE?
1042	002412	001005			BNE LCONM ; BRANCH IF YES
1043	002414	052767	070000	015042	BIS #70000, FLAG
1044	002422	000167	000726		JMP ADTST
1045					; ENTER OPERATOR CONVERSATION MODE
1046	002426				LCONM:
1047	002426	004567	176650		JSR RS, PRNTFS ; FORCE PRINT THE MESSAGE
1048	002432	016630			SPECMES
1049	002434	016767	015050	177120	MOV SWRDCT, TTY
1050	002442	004767	176732		JSR PC, PRINTB ; FORCE TYPE LOCATION - SUPPRESS ZEROS
1051	002446	004567	176630		JSR RS, PRNTFS ; FORCE PRINT THE MESSAGE
1052	002452	016667			CON1 ; ASK ABOUT DATA TEST ONLY
1053	002454	004767	177224		JSR PC, READS ; INPUT MESSAGE
1054	002460	122767	000131	177326	CMPB #131, INPUTS ; TEST FOR YES
1055	002466	001003			BNE .+10 ; BRANCH IF NO
1056	002470	052767	002000	014766	BIS #B10, FLAG ; SET DATA TEST ONLY FLAG
1057	002476	004567	176600		JSR RS, PRNTFS ; FORCE PRINT THE MESSAGE
1058	002502	016722			CON2 ; ASK ABOUT MULTI DRIVE MODE
1059	002504	004767	177174		JSR PC, READS ; INPUT MESSAGE
1060	002510	122767	000131	177276	CMPB #131, INPUTS ; TEST FOR YES
1061	002516	001040			BNE DATTES ; BRANCH IF NO
1062	002520	052767	004000	014736	BIS #B11, FLAG ; SET MULTI UNIT FLAG
1063	002526				DSKDR:
1064	002526	004567	176550		JSR RS, PRNTFS ; FORCE PRINT THE MESSAGE
1065	002532	016756			CON3 ; GET NO. OF UNITS
1066	002534	004767	177144		JSR PC, READS ; INPUT MESSAGE
1067	002540	004767	177316		JSR PC, PACKS ; CONVERT INPUT TO A NUMBER
1068	002544	005767	177564		TST NUMS ; IS IT ZERO
1069	002550	001766			BEQ DSKDR
1070	002552	162767	000001	177554	SUB #1, NUMS
1071	002560	022767	000010	177546	CMP #10, NUMS ; IS NO. TOO HIGH
1072	002566	101757			BLOS DSKDR
1073	002570	016767	177540	014722	MOV NUMS, DSKNOR ; SAVE HIGHEST UNIT NO.
1074	002576	042767	177770	014714	BIC #177770, DSKNOR
1075	002604	000241			CLC
1076	002606	006167	014706		ROL DSKNOR
1077	002612	006167	014702		ROL DSKNOR
1078	002616	000423			BR ASKWC
1079	002620				DATTES:
1080	002620	004567	176456		JSR RS, PRNTFS ; FORCE PRINT THE MESSAGE
1081	002624	017020			CON4 ; ASK UNIT NUMBER
1082	002626	004767	177052		JSR PC, READS ; INPUT MESSAGE
1083	002632	004767	177224		JSR PC, PACKS ; CONVERT INPUT TO A NUMBER
1084	002636	022767	000010	177470	CMP #10, NUMS ; IS NO = OR > 10
1085	002644	101765			BLOS DATTES ; NO
1086	002646	000241			CLC
1087	002650	006167	177460		ROL NUMS

Line No.	Code	Address	Value	Label	Op Code	Comments
1088	002654	006167	177454		RUL	NUMS
1089	002660	056767	177450	014576	BIS	NUMS, FLAG ;SAVE UNIT UNDER TEST
1090	002666				ASKWC:	
1091	002666	004567	176410		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1092	002672	017037			CONS	;ASK ABOUT OPTIONAL WORD COUNT
1093	002674	004767	177004		JSR	PC, READS ;INPUT MESSAGE
1094	002700	122767	000131	177106	CMPB	#131, INPUTS ;TEST FOR YES
1095	002706	001031			BNE	TKSR ;ASK ABOUT OPTIONAL DAR
1096	002710				WCCON:	
1097	002710	004567	176366		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1098	002714	017077			CON6	;ASK LENGTH OF WC
1099	002716	004767	176762		JSR	PC, READS ;INPUT MESSAGE
1100	002722	004767	177134		JSR	PC, PACKS ;CONVERT INPUT TO A NUMBER
1101	002726	005767	177402		TST	NUMS
1102	002732	001766			BEQ	WCCON
1103	002734	016767	014550	014600	MOV	SWRDC?, WORK
1104	002742	005267	014574		INC	WORK
1105	002746	026767	014570	177360	CMP	WORK, NUMS ;IS NO. GREATER THAN AVAILABLE CORE?
1106	002754	101755			BLOS	WCCON ;YES ASK FOR COUNT AGAIN
1107	002756	016767	177352	014524	MOV	NUMS, SWRDC? ;OPERATING WORD COUNT
1108	002764	016767	014520	014502	MOV	SWRDC?, WRDC?
1109	002772				TKSR:	
1110	002772	004567	176304		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1111	002776	017150			CON7A	;ASK ABOUT DISK ADDR
1112	003000	004767	176700		JSR	PC, READS ;INPUT MESSAGE
1113	003004	122767	000131	177002	CMPB	#131, INPUTS ;WILL OPERATOR SUPPLY ADDR?
1114	003012	001055			BNE	OPPAT ;BRANCH IF NO
1115	003014	052767	000040	014442	BIS	#85, FLAG
1116	003022				OPDAR:	
1117	003022	004567	176254		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1118	003026	017257			CON7C	;GET CYLINDER ADDR
1119	003030	004767	176650		JSR	PC, READS ;INPUT MESSAGE
1120	003034	004767	177022		JSR	PC, PACKS ;CONVERT INPUT TO A NUMBER
1121	003040	022767	000313	177266	CMP	#313, NUMS ;IS CYLINDER LEGAL
1122	003046	101765			BLOS	OPDA1
1123	003050	016767	177260	014410	MOV	NUMS, SCYL ;SAVE ADDR
1124	003056				OPDA1:	
1125	003056	004567	176220		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1126	003062	017127			CON7	;GET HEAD ADDR
1127	003064	004767	176614		JSR	PC, READS ;INPUT MESSAGE
1128	003070	004767	176766		JSR	PC, PACKS ;CONVERT INPUT TO A NUMBER
1129	003074	022767	000024	177232	CMP	#24, NUMS
1130	003102	101765			BLOS	OPDA1 ;BRANCH IF HEAD ADDR TOO HIGH
1131	003104	016767	177224	014356	MOV	NUMS, SHED ;SAVE ADDR
1132	003112				OPDA2:	
1133	003112	004567	176164		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1134	003116	017234			CON7B	;GET SECTOR ADDR
1135	003120	004767	176560		JSR	PC, READS ;INPUT MESSAGE
1136	003124	004767	176732		JSR	PC, PACKS ;CONVERT INPUT TO A NUMBER
1137	003130	022767	000012	177176	CMP	#12, NUMS ;IS SECTOR ADDR TOO HIGH?
1138	003136	101765			BLOS	OPDA2
1139	003140	016767	177170	014324	MOV	NUMS, SSEC ;SAVE ADDR
1140						
1141	003146				OPPAT:	
1142	003146	004567	176130		JSR	RS, PRNTFS ;FORCE PRINT THE MESSAGE
1143	003152	017304			CON8	;ASK ABOUT DATA PATTERNS

1144	003154	004767	176524			JSR	PC, READS	: INPUT MESSAGE
1145	003160	004767	176676			JSR	PC, PACKS	: CONVERT INPUT TO A NUMBER
1146	003164	022767	000020	177142		CMP	#20, NUMS	: TEST FOR CORRECT NO
1147	003172	101765				BLOS	OPPAT	: ASK AGAIN
1148	003174	022767	000017	177132		CMP	#17, NUMS	
1149	003202	001411				BEQ	OPWRT	: DATA PATTERN UNDER PROGRAM CONTROL
1150	003204	052767	100000	014252		BIS	#B15, FLAG	: SET PROGRAM FLAG
1151	003212	016767	177116	014264		MOV	NUMS, PATNU	: OPERATOR WANTS TO SELECT DATA
1152	003220	000241				CLC		
1153	003222	006167	014256			ROL	PATNU	
1154	003226	042767	070000	014230	OPWRT:	BIC	#70000, FLAG	: CLEAR OP MODE BITS IN FLAG
1155	003234	004567	176042			JSR	RS, PRNTFS	: FORCE PRINT THE MESSAGE
1156	003240	017341				CON9		: ASK ABOUT WRITE
1157	003242	004767	176436			JSR	PC, READS	: INPUT MESSAGE
1158	003246	122767	000131	176540		CMPB	#131, INPUTS	: TEST FOR YES
1159	003254	001003				BNE	OPRD	: ASK ABOUT WRITE CHECK
1160	003256	052767	040000	014200		BIS	#B14, FLAG	: YES SET FLAG BIT
1161	003264				OPRD:			
1162	003264	004567	176012			JSR	RS, PRNTFS	: FORCE PRINT THE MESSAGE
1163	003270	017411				CON11		: ASK ABOUT READ
1164	003272	004767	176406			JSR	PC, READS	: INPUT MESSAGE
1165	003276	122767	000131	176510		CMPB	#131, INPUTS	: TEST FOR YES ANSWER
1166	003304	001003				BNE	OPWCK	
1167	003306	052767	010000	014150		BIS	#B12, FLAG	: SET FLAG TO READ
1168	003314				OPWCK:			
1169	003314	004567	175762			JSR	RS, PRNTFS	: FORCE PRINT THE MESSAGE
1170	003320	017362				CON10		: ASK ABOUT WRITE CHECK
1171	003322	004767	176356			JSR	PC, READS	: INPUT MESSAGE
1172	003326	122767	000131	176460		CMPB	#131, INPUTS	
1173	003334	001003				BNE	CHKMOD	
1174	003336	052767	020000	014120		BIS	#B13, FLAG	: SET WRITE CHECK FLAG
1175	003344	032767	070000	014112	CHKMOD:	BIT	#70000, FLAG	: MAKE SURE SOME OPERATION WAS SELECTED
1176	003352	001725				BEQ	OPWRT	
1177								
1178								
1179	003354	005737	000042		ADTST:	TST	#42	: UNDER MONITOR CONTROL?
1180	003360	001444				BEQ	15	: BRANCH IF NO
1181	003362	005067	014132			CLR	DSKNOR	
1182	003366	012777	000001	014044	3S:	MOV	#1, JRPCS	: CLEAR THE RP11C
1183	003374	116777	014120	014040		MOVB	DSKNOR, JRPCS1	: SELECT THE DRIVE
1184	003402	005777	014052			TST	JRPDS	: IS THE UNIT READY?
1185	003406	100003				BPL	2S	: BRANCH IF NO
1186	003410	005267	014104			INC	DSKNOR	: UPDATE UNIT NUMBER
1187	003414	000764				BR	3S	
1188	003416	005367	014076		2S:	DEC	DSKNOR	: DSKNOR = NUMBER OF UNITS
1189	003422	000241				CLC		
1190	003424	006167	014070			ROL	DSKNOR	
1191	003430	006167	014064			ROL	DSKNOR	
1192	003434	052767	004000	014022		BIS	#B11, FLAG	: SET MULTI DRIVE FLAG
1193	003442	005767	014052			TST	DSKNOR	: WERE ANY UNITS AVAILABLE?
1194	003446	100011				BPL	15	: BRANCH IF YES
1195	003450	004567	175610			JSR	RS, PRINTS	: PRINT MESSAGE
1196	003454	016535				MES20		
1197	003456	013701	000042			MOV	#42, R1	: ABORT - NO UNITS AVAILABLE
1198	003462	005067	174354			CLR	42	: SET ABORT FLAG
1199	003466	000167	006366			JMP	MEXIT	

```

1200 003472 032767 004000 013764 1$: BIT #B11,FLAG ;ARE WE IN MULTI DRIVE MODE?
1201 003500 001422 BEQ EXMFLG ;BRANCH IF NO
1202 003502 004567 175574 JSR RS,PRINTS ;FORCE PRINT THE MESSAGE
1203 003506 016370 MES11
1204 003510 016767 013750 014014 MOV FLAG,ACNVX ;TELL OPERATOR THE UNIT UNDER TEST
1205 003516 006067 014010 ROR ACNVX
1206 003522 006067 014004 ROR ACNVX
1207 003526 042767 177770 013776 BIC #177770,ACNVX
1208 003534 016767 013772 176020 MOV ACNVX,TTY
1209 003542 004767 175632 JSR PC,PRINTB ;FORCE TYPE LOCATION - SUPPRESS ZEROS
1210 003546 032737 000010 177570 EXMFLG: BIT #B3,#SWR ;RUN SELECTED TEST?
1211 003554 001410 BEQ 1$ ;BRANCH IF NO
1212 003556 013700 177570 MOV #SWR,RO ;GET SWITCH SETTINGS
1213 003562 042700 177770 BIC #177770,RO
1214 003566 000241 CLC
1215 003570 006100 ROL RO
1216 003572 000170 003612 JMP #TSTTBL(RO) ;GO TO SELECTED TEST
1217 003576 032767 002000 013660 1$: BIT #B10,FLAG ;DATA TEST ONLY?
1218 003604 001412 BEQ ADT1 ;NO
1219 003606 000167 004520 JMP DATAT ;DO DATA TEST

```

```

1220
1221 003612 003632 TSTTBL: ADT1
1222 003614 004312 ADT2
1223 003616 005370 ADT3
1224 003620 006024 WRCK
1225 003622 007220 MEMTST
1226 003624 010332 DATAT
1227 003626 011166 RANEX
1228 003630 012074 PFTST

```

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

:IN THIS TEST THE PROGRAM SEEKS FROM 0 TO N AND THEN BACK
:TO 0. N STARTS AT ZERO THEN INCREMENTS TO 1 AND UP THRU 312
:DONE IS TIMED OUT, SELECTED UNIT CYLINDER ADDRESS IS TESTED, SEEK UNDERWAY
:IS CHECK, AND THE ATTENTION FLAG IS TESTED.

```

1240 003632 005067 013700 ADT1: CLR TESTNO
1241 003636 004567 175422 JSR RS,PRINTS ;PRINT MESSAGE
1242 003642 016225 MES6
1243 003644 016767 013666 175710 MOV TESTNO,TTY
1244 003652 004767 175500 JSR PC,PRINTS ;TYPE LOCATION-SUPPRESS ZEROS
1245 003656 012737 004300 000254 RADT1: MOV #INTCK,#VECTOR ;SET UP DISK VECTOR
1246 003664 012737 000340 000256 MOV #340,#STATUS
1247 003672 004567 007752 JSR RS,DSKNOS ;SELECT UNIT
1248 003676 005067 013574 CLR CYLINDER
1249 003702 005067 013640 CLR WORK2 ;CYLINDER COUNTER
1250 003706 005067 013636 CLR WORK3 ;POINTER
1251 003712 012737 000200 177776 MOV #PRI4,#PSW ;ALLOW INTERRUPTS
1252 003720 047777 013534 013532 9$: BIC #RPDS,#RPDS ;CLEAR ATTENTION BITS
1253 003726 116777 013544 013514 MOVB CYLINDER,#RPCA ;SET CYLINDER REGISTER
1254 003734 052777 020011 013476 BIS #20011,#RPCS ;SEEK AND ENABLE ATTN INTERRUPT
1255 003742 005067 000342 CLR INTFLG ;CLEAR INTERRUPT FLAG

```

1256	003746	012700	000025			MOV	#25,RO	
1257	003752	005300		1\$:		DEC	RO	;DELAY FOR DONE TO SET
1258	003754	001376				BNE	1\$	
1259	003756	105777	013456			TSTB	BRPCS	;TEST FOR DONE
1260	003762	100402				BMI	2\$;BRANCH DONE SET
1261	003764	104400				HLT		;DONE DID NOT SET AFTER SEEK
1262	003766	000471				BR	8\$	
1263	003770	005767	013552	2\$:		TST	WORK2	;DON'T TEST SEEK UNDERWAY
1264	003774	001406				BEQ	3\$;IF FIRST TIME THRU
1265	003776	032777	002000	013454		BIT	#B10,BRPDS	;DID SEEK UNDERWAY SET?
1266	004004	001002				BNE	3\$;BRANCH IF YES
1267	004006	104400				HLT		;SEEK UNDERWAY DID NOT SET
1268	004010	000460				BR	8\$	
1269	004012	005000		3\$:		CLR	RO	
1270	004014	005200		5\$:		INC	RO	;TIMEOUT UNIT READY
1271	004016	005777	013436			TST	BRPDS	;IS UNIT READY?
1272	004022	100414				BMI	6\$;BRANCH IF YES
1273	004024	005237	017554			INC	#CYLA	
1274	004030	005337	017554			DEC	#CYLA	
1275	004034	005237	017554			INC	#CYLA	
1276	004040	005337	017554			DEC	#CYLA	
1277	004044	005700				TST	RO	;TIMEOUT?
1278	004046	001362				BNE	5\$;BRANCH IF NO
1279	004050	104400				HLT		;READY DID NOT SET AFTER SEEK
1280	004052	000437				BR	8\$	
1281	004054	005767	000230	6\$:		TST	INTFLG	;DID INTERRUPT OCCUR?
1282	004060	001002				BNE	12\$;BRANCH IF YES
1283	004062	104400				HLT		;INTERRUPT DID NOT OCCUR ON ATTENTION BIT
1284	004064	000432				BR	8\$	
1285	004066	004767	007764	12\$:		JSR	PC,GATTN	;DETERMINE ATTENTION BIT
1286	004072	036777	010010	013360		BIT	ATTN,BRPDS	;IS ATTENTION BIT SET?
1287	004100	001002				BNE	7\$;BRANCH IF YES
1288	004102	104400				HLT		;ATTENTION BIT DID NOT SET
1289	004104	000422				BR	8\$	
1290	004106	126777	013364	013346	7\$:	CMPB	CYLINDER,BRPCA1	;IS SELECTED CYLINDER ADDRESS CORRECT?
1291	004114	001412				BEQ	11\$	
1292	004116	016767	013354	010426		MOV	CYLINDER,EXPS	;EXPECTED RESULTS
1293	004124	005067	010424			CLR	RECS	
1294	004130	117767	013326	010416		MOVB	BRPCA1,RECS	;RECEIVED RESULTS
1295	004136	104401				HLT	+1	;CONTENTS OF SELECTED CYLINDER ADDR REGISTER INCORRECT
1296	004140	000404				BR	8\$	
1297	004142	005777	013272	11\$:		TST	BRPCS	;ANY DEVICE ERRORS
1298	004146	100001				BPL	8\$;BRANCH IF NO
1299	004150	104400				HLT		;DEVICE ERROR AFTER SEEK OPERATION
1300	004152	032777	004000	013300	8\$:	BIT	#B11,BRPDS	;SEEK INCOMPLETE?
1301	004160	001411				BEQ	4\$;BRANCH IF NO
1302	004162	112777	000015	013250		MOVB	#15,BRPCS	;ISSUE HOME COMMAND
1303	004170	105777	013244			TSTB	BRPCS	;WAIT FOR DONE
1304	004174	100375				BPL	-4	
1305	004176	005777	013256			TST	BRPDS	;WAIT FOR UNIT READY
1306	004202	100375				BPL	-4	
1307	004204	012767	003720	174570	4\$:	MOV	#9\$,LAD	;SET UP LOOP
1308	004212	104000				SCOPE		
1309	004214	005767	013330			TST	WORK3	;SEEK CYLINDER ZERO?
1310	004220	100411				BMI	10\$;BRANCH IF YES
1311	004222	005267	013320			INC	WORK2	;UPDATE CYLINDER

1312	004226	016767	013314	013242		MOV	WORK2,CYLINDER	
1313	004234	052767	100000	013306		BIS	#B15,WORK3	;SET SEEK ZERO FLAG
1314	004242	000626				BR	9S	
1315	004244	005067	013226		10S:	CLR	CYLINDER	
1316	004250	005067	013274			CLR	WORK3	;CLEAR SEEK ZERO FLAG
1317	004254	022767	000312	013264		CMP	#312,WORK2	;HAS LAST CYLINDER BEEN REACHED?
1318	004262	001216				BNE	9S	;BRANCH IF NO
1319	004264	032737	002000	177570		BIT	#B10,0#SWR	;REPEAT TEST
1320	004272	001407				BEQ	ADT2	;NO-GO TC NEXT
1321	004274	000167	177356			JMP	RADT1	;YES
1322								
1323								
1324	004300	012767	000001	000002	INTCK:	MOV	#1,INTFLG	;SET INTERRUPT FLAG
1325	004306	000002				RTI		
1326	004310	000000			INTFLG:	0		

1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382

.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.

```

1340 004312 012767 000001 013216 ADT2:  MOV      #1,TESTNO
1341 004320 004567 174740          JSR      RS,PRINTS      ;PRINT MESSAGE
1342 004324 016225          MES6
1343 004326 016767 013204 175226  MOV      TESTNO,TTY
1344 004334 004767 175016          JSR      PC,PRINTS      ;TYPE LOCATION-SUPRESS ZEROS
1345 004340 004567 007304          RADT2: JSR      RS,DSKNOS      ;SELECT THE DRIVE
1346 004344 052777 000015 013066 1$:     BIS      #15,DRPCS      ;SEEK HOME
1347 004352 012700 000025          MOV      #25,R0
1348 004356 005300          2$:     DEC      R0          ;GIVE DONE A CHANCE TO SET
1349 004360 001376          BNE     2$
1350 004362 105777 013052          TSTB    DRPCS          ;IS DONE SET?
1351 004366 100402          BMI     3$          ;YES-BRANCH
1352 004370 104400          HLT
1353 004372 000425          BR      6$          ;DONE DID NOT SET AFTER A SEEK HOME
1354 004374 005000          3$:     CLR      R0          ;CHECK FOR LOOPING
1355 004376 005200          5$:     INC      R0
1356 004400 005777 013054          TST     DRPDS          ;IS UNIT READY?
1357 004404 100414          BMI     4$          ;YES BRANCH
1358 004406 005237 017554          INC     @#CYLA
1359 004412 005337 017554          DEC     @#CYLA
1360 004416 005237 017554          INC     @#CYLA
1361 004422 005337 017554          DEC     @#CYLA
1362 004426 005700          TST     R0          ;HAS UNIT TIMED OUT
1363 004430 001362          BNE     5$          ;NO-BRANCH
1364 004432 104400          HLT
1365 004434 000404          BR      6$          ;READY DID NOT SET AFTER HOME SEEK
1366 004436 005777 012776          4$:     TST     DRPCS      ;CHECK FOR LOOPING
1367 004442 100001          BPL     6$          ;ANY ERRORS?
1368 004444 104400          HLT          ;NO-BRANCH
1369 004446 012767 004344 174326 6$:     MOV      #1$,LAD      ;DRIVE ERRORS AFTER HOME SEEK
1370 004454 104000          SCOPE
1371 004456 012767 005000 013010  MOV      #5000,WRDCT   ;SETUP WORD COUNT FOR 10 SECTORS
1372 004464 012767 017556 013014  MOV      #OUTBUF,BUF   ;SETUP OUTPUT BUFFER ADDR
1373 004472 005067 013002          CLR     DMA
1374 004476 005067 012774          CLR     CYLINDER
1375 004502 012700 017556          SEABUF: MOV      #OUTBUF,R0  ;GET BUFFER STARTING ADDR
1376 004506 012701 000400          21$:    MOV      #400,R1   ;SECTOR COUNT
1377 004512 016720 012760          MOV     CYLINDER,(R0)+ ;GENERATE PATTERN SO THAT THE
1378 004516 005301          DEC     R1            ;THE FIRST WORD OF EACH SECTOR
1379 004520 016720 012754          1$:     MOV     DMA,(R0)+  ;EQUALS THE CYLINDER ADDR AND
1380 004524 005301          DEC     R1            ;THE REMAINDER EQUALS THE HEAD AND
1381 004526 001374          BNE     1$           ;SECTOR ADDR
1382 004530 122767 000011 012742  CMPB    #11,DMA
  
```

1383	004536	001403				BEQ	22\$		
1384	004540	005267	012734			INC	DMA		;UPDATE SECTOR COUNT
1385	004544	000760				BR	21\$		
1386	004546	105067	012726		22\$:	CLRB	DMA		
1387	004552	004567	005742		4\$:	JSR	RS,FUNCT		;WRITE TEN SECTORS
1388	004556	000003			.WORD	3			
1389	004560	005000				CLR	RO		
1390	004562	005200			3\$:	INC	RO		
1391	004564	105777	012650			TSTB	DRPCS		; IS DONE SET?
1392	004570	100404				BMI	2\$; YES BRANCH
1393	004572	005700				TST	RO		; TEST FOR TIMEOUT
1394	004574	001372				BNE	3\$; BRANCH IF NO
1395	004576	104400				HLT			; DONE DID NOT SET AFTER WRITE
1396	004600	000503				BR	5\$		
1397	004602	005777	012632		2\$:	TST	DRPCS		; ANY DEVICE ERRORS?
1398	004606	100002				BPL	6\$; BRANCH IF NO
1399	004610	104400				HLT			; RP11C STATUS ERROR AFTER WRITE
1400	004612	000476				BR	5\$		
1401	004614	005067	012722		6\$:	CLR	WORK		; INCREMENT FLAG
1402	004620	017767	012626	007726		MOV	DRPDA,RECS		; GET DISK ADDR
1403	004626	042767	177760	007720		BIC	#177760,RECS		; SAVE SECTOR ADDR
1404	004634	005767	007714			TST	RECS		
1405	004640	001404				BEQ	9\$; BRANCH IF SECTOR = ZERO
1406	004642	005067	007704			CLR	EXPS		
1407	004646	104401				HLT	+1		; SECTOR ADDR IN RPDA DID NOT UPDATE
1408	004650	000457				BR	5\$; PROPERLY AFTER A TEN SECTOR WRITE
1409	004652	117767	012576	007674	9\$:	MOVB	DRPDA1,RECS		; GET THE HEAD ADDR
1410	004660	116767	012615	007664		MOVB	DMA+1,EXPS		; SECTOR ADDR OUTPUTTED
1411	004666	122767	000023	007656		CMPB	#23,EXPS		; DID WE OUTPUT HEAD 23?
1412	004674	001005				BNE	7\$; BRANCH IF NO
1413	004676	005067	007650			CLR	EXPS		; RESET HEAD ADDR
1414	004702	010667	012634			MOV	SP,WORK		; SET INCREMENT FLAG
1415	004706	000402				BR	8\$		
1416	004710	005267	007636		7\$:	INC	EXPS		
1417	004714	126767	007632	007632	8\$:	CMPB	EXPS,RECS		; IS DISK HEAD ADDR CORRECT?
1418	004722	001402				BEQ	12\$; BRANCH IF YES
1419	004724	104401				HLT	+1		; HEAD ADDR IN RPDA WAS INCORRECT
1420	004726	000430				BR	5\$; AFTER TEN SECTOR WRITE
1421	004730	005067	007620		12\$:	CLR	RECS		
1422	004734	117767	012510	007612		MOVB	DRPCA,RECS		; GET DISK CYLINDER ADDR
1423	004742	016767	012530	007602		MOV	CYLINDER,EXPS		
1424	004750	005767	012566			TST	WORK		; IS INCREMENT FLAG SET?
1425	004754	001410				BEQ	13\$; BRANCH IF NO
1426	004756	005267	007570			INC	EXPS		
1427	004762	022767	000313	007562		CMP	#313,EXPS		; WAS IT LAST CYLINDER?
1428	004770	001002				BNE	13\$; BRANCH IF NO
1429	004772	005067	007554			CLR	EXPS		
1430	004776	026767	007550	007550	13\$:	CMP	EXPS,RECS		; IS DISK CYLINDER ADDR CORRECT?
1431	005004	001401				BEQ	5\$; BRANCH IF YES
1432	005006	104401				HLT	+1		; CYLINDER ADDR IN RPDA IS NOT
1433									; CORRECT AFTER TEN SECTOR WRITE
1434	005010	032777	004000	012442	5\$:	BIT	#B11,DRPDS		; SEEK INCOMPLETE?
1435	005016	001411				BEQ	10\$; BRANCH IF NO
1436	005020	112777	000015	012412		MOVB	#15,DRPCS		; ISSUE HOME COMMAND
1437	005026	105777	012406			TSTB	DRPCS		; WAIT FOR DONE
1438	005032	100375				BPL	.-4		

```

1439 005034 005777 012420          TST      ARPDS          ;WAIT FOR UNIT READY
1440 005040 100375          BPL      .-4
1441 005042 012767 004552 173732 10$:    MOV      #4$,LAD        ;SETUP LOOP ADDR
1442 005050 104000          SCOPE
1443 005052 004767 006162          JSR      PC,DISBUF      ;SETUP NEXT DISK ADDR
1444 005056 000611          BR       SEABUF         ;WRITE NEXT SECTOR
1445 005060 012767 005000 012406    MOV      #5000,WRDCT    ;RESTORE WORD COUNT
1446 005066 052777 000015 012344    BIS      #15,ARPCS     ;SEEK HOME
1447 005074 105777 012340          TSTB     ARPDS
1448 005100 100375          BPL      .-4           ;WAIT FOR DONE AFTER SEEK HOME
1449 005102 005777 012352          TST      ARPDS
1450 005106 100375          BPL      .-4           ;WAIT FOR DRIVE READY AFTER SEEK HOME
1451 005110 012700 017556    RDSECT: MOV      #OUTBUF,RO
1452 005114 012701 005000          MOV      #5000,R1
1453 005120 005020 23$:      CLR      (RO)+         ;CLEAR THE BUFFER
1454 005122 005301          DEC      R1
1455 005124 001375          BNE      23$
1456 005126 004567 005366          JSR      R5,FUNCT      ;READ TEN SECTORS
1457 005132 000005    .WORD    5
1458 005134 105777 012300          TSTB     ARPDS
1459 005140 100375          BPL      .-4           ;WAIT FOR DONE AFTER READ
1460 005142 005777 012272          TST      ARPDS        ;ANY ERRORS?
1461 005146 100006          BPL      ADHGT         ;BRANCH NO ERRORS
1462 005150 104400          HLT
1463 005152 032777 040000 012260    BIT      #B14,ARPCS    ;STATUS ERROR AFTER A READ
1464 005160 001401          BEQ      ADHGT         ;WAS IT A DATA ERROR?
1465 005162 000446          BR       ADTER1
1466 005164 012700 017556    ADHGT:  MOV      #OUTBUF,RO
1467 005170 012701 000400    ADHGT1: MOV      #400,R1
1468 005174 026710 012276          CMP      CYLINDER,(0) ; IS CYLINDER WORD CORRECT?
1469 005200 001017          BNE      ADERC        ;BRANCH IF NO
1470 005202 005720          TST      (0)+
1471 005204 005301          DEC      R1
1472 005206 026710 012266    SANHT:  CMP      DMA,(0) ; IS HEAD-SECTOR WORD CORRECT?
1473 005212 001016          BNE      ADERR        ;BRANCH IF NO
1474 005214 005720          TST      (0)+
1475 005216 005301          DEC      R1
1476 005220 001372          BNE      SANHT
1477 005222 122767 000011 012250    CMPB     #11,DMA
1478 005230 001423          BEQ      ADTER1
1479 005232 005267 012242          INC      DMA
1480 005236 000754          BR       ADHGT1
1481 005240 016767 012232 007304    ADERC:  MOV      CYLINDER,EXPS ; CORRECT DATA/ADDRESS
1482 005246 000403          BR       ADERC1
1483 005250 016767 012224 007274    ADERR:  MOV      DMA,EXPS ; CORRECT DATA/ADDRESS
1484 005256 011067 007272          ADERC1: MOV      (0),RECS ; INCORRECT DATA
1485 005262 104401          HLT      +1           ; DATA COMPARE ERROR
1486 005264 022701 000400          CMP      #400,R1      ; WAS FIRST WORD INCORRECT?
1487 005270 001003          BNE      ADTER1      ; BRANCH IF NO
1488 005272 004567 173766          JSR      R5,PRINT$    ; PRINT MESSAGE
1489 005276 016404          MES12
1490 005300 105067 012174    ADTER1: CLRB     DMA
1491 005304 032777 004000 012146    BIT      #B11,ARPDS   ; SEEK INCOMPLETE?
1492 005312 001411          BEQ      1$           ; BRANCH IF NO
1493 005314 112777 000015 012116    MOVB     #15,ARPCS    ; ISSUE HOME COMMAND
1494 005322 105777 012112          TSTB     ARPDS        ; WAIT FOR DONE

```

K03

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 36
DZRPFB.P11 ***** TEST 1 *****

1495	005326	100375			BPL	.-4	
1496	005330	005777	012124		TST	@RPDS	;WAIT FOR UNIT READY
1497	005334	100375			BPL	.-4	
1498	005336	012767	005110	173436	MOV	#RDSECT,LAD	;SETUP LOOP
1499	005344	104000			SCOPE		
1500	005346	004767	005666		JSR	PC,DISBUF	;SETUP NEXT DISK ADDRESS
1501	005352	000656			BR	RDSECT	;CHECK NEXT SECTOR
1502	005354	032737	002000	177570	BIT	#B10,@#SWR	;LOOP ON TEST?
1503	005362	001402			BEQ	ADT3	;BRANCH IF NO
1504	005364	000167	176750		JMP	RADT2	

```

1505 .SBTTL ***** TEST 2 *****
1506
1507 ;WRITE THE FIRST WORD OF EACH CYLINDER WITH THE CYLINDER ADDRESS.
1508 ;THEN SEEK FROM 312 TO 0. THEN SEEK TO 311 AND BACK TO 1
1509 ;UNTIL THE NUMBERS CROSS. AFTER EACH SEEK VERIFY THE POSITION
1510 ;BY READING THE FIRST WORD ON THE CYLINDER AND COMPARING
1511 ;AGAINST CYLINDER REQUESTED.
1512
1513 005370 012767 000002 012140 ADT3: MOV #2,TESTNO
1514 005376 004567 173662 JSR RS,PRINTS ;PRINT MESSAGE
1515 005402 016225 MES6
1516 005404 016767 012126 174150 MOV TESTNO,TTY
1517 005412 004767 173740 JSR PC,PRINTS ;TYPE LOCATION-SUPPRESS ZEROS
1518 005416 004767 006300 RADT3: JSR PC,INIT ;INITIALIZE VECTORS
1519 005422 004567 006222 JSR RS,DSKNOS ;SELECT UNIT
1520 005426 005067 012046 CLR DMA
1521 005432 005067 012040 CLR CYLINDER
1522 005436 012767 000001 012030 MOV #1,WRDCT ;SET UP WORD COUNT
1523 005444 012767 017556 012034 MOV #OUTBUF,BUF ;SETUP BUFFER ADDR
1524 005452 016737 012020 017556 WRCYL: MOV CYLINDER,#OUTBUF ;INSERT PATTERN
1525 005460 004567 005034 JSR RS,FUNCT ;WRITE PATTERN ON FIRST SECTOR
1526 005464 000003 .WORD 3 ;OF CYLINDER
1527 005466 105777 011746 TSTB @RPCS ;WAIT FOR DONE
1528 005472 100375 BPL -4
1529 005474 005777 011740 TST @RPCS ;AND ERRORS?
1530 005500 100002 BPL 1$ ;BRANCH IF NO
1531 005502 104400 HLT ;DEVICE ERROR WHILE WRITING
1532 005504 000407 BR 2$
1533 005506 022767 000312 011762 1$: CMP #312,CYLINDER ;ALL CYLINDERS WRITTEN?
1534 005514 001403 BEQ 2$ ;BRANCH IF YES
1535 005516 005267 011754 INC CYLINDER
1536 005522 000753 BR WRCYL
1537 005524 032777 004000 011726 2$: BIT #B11,@RPS ;SEEK INCOMPLETE?
1538 005532 001411 BEQ 3$ ;BRANCH IF NO
1539 005534 112777 000015 011676 MOVB #15,@RPS ;ISSUE HOME COMMAND
1540 005542 105777 011672 TSTB @RPS ;WAIT FOR DONE
1541 005546 100375 BPL -4
1542 005550 005777 011704 TST @RPS ;WAIT FOR UNIT READY
1543 005554 100375 BPL -4
1544 005556 012767 005452 173216 3$: MOV #WRCYL,LAD ;SETUP UP LOOP
1545 005564 104000 SCOPE
1546 005566 005067 011704 CLR CYLINDER
1547 005572 012767 000311 011746 MOV #311,WORK2
1548 005600 005067 011744 CLR WORK3
1549 005604 005067 011742 CLR WORK4 ;INC - DEC FLAG
1550 005610 016767 011712 011670 ADT32: MOV INBUF,BUF
1551 005616 004567 004676 JSR RS,FUNCT ;READ THE FIRST WORD OF THE
1552 005622 000005 .WORD 5 ;CYLINDER
1553 005624 105777 011610 TSTB @RPCS ;WAIT FOR DONE AFTER READ
1554 005630 100375 BPL -4
1555 005632 005777 011602 TST @RPCS ;ANY ERRORS?
1556 005636 100002 BPL 3$ ;BRANCH IF NO
1557 005640 104400 HLT ;ERROR AFTER READING ONE WORD
1558 005642 000413 BR 5$
1559 005644 027767 011656 011624 3$: CMP @INBUF,CYLINDER ;COMPARE DATA READ AGAINST CYLINDER
1560 005652 001407 BEQ 5$ ;BRANCH IF EQUAL
  
```

1561	005654	016767	011616	006670		MOV	CYLINDER,EXPS	; CORRECT DATA
1562	005662	017767	011640	006664		MOV	INBUF,RECS	; INCORRECT DATA
1563	005670	104401				HLT	+1	; DATA COMPARE ERROR-PROBABLY WENT
1564								; TO THE WRONG CYLINDER
1565	005672	032777	004000	011560	5\$:	BIT	#B11,ARPCS	; SEEK INCOMPLETE?
1566	005700	001411				BEQ	2\$; BRANCH IF NO
1567	005702	112777	000015	011530		MOVB	#15,ARPCS	; ISSUE HOME COMMAND
1568	005710	105777	011524			TSTB	ARPCS	; WAIT FOR DONE
1569	005714	100375				BPL	-4	
1570	005716	005777	011536			TST	ARPCS	; WAIT FOR UNIT READY
1571	005722	100375				BPL	-4	
1572	005724	012767	005610	173050	2\$:	MOV	#ADT32,LAD	; SETUP LOOP
1573	005732	104000				SCOPE		
1574	005734	005767	011612			TST	WORK4	; INC - DEC FLAG
1575	005740	100411				BMI	1\$	
1576	005742	005267	011602			INC	WORK3	; UPDATE LOW COUNT
1577	005746	016767	011574	011522		MOV	WORK2,CYLINDER	
1578	005754	052767	100000	011570		BIS	#B15,WORK4	; SET DECREMENT FLAG
1579	005762	000712				BR	ADT32	
1580	005764	005367	011556		1\$:	DEC	WORK2	; DECREMENT HIGH COUNT
1581	005770	005067	011556			CLR	WORK4	; CLEAR FLAG
1582	005774	016767	011550	011474		MOV	WORK3,CYLINDER	
1583	006002	005767	011540			TST	WORK2	; DONE YET
1584	006006	001300				BNE	ADT32	; BRANCH-NO
1585	006010	032737	002000	177570		BIT	#B10,#SWR	; LOOP ON TEST?
1586	006016	001402				BEQ	WRCK	; NO
1587	006020	000167	177372			JMP	RADT3	; YES

1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643

006024 012767 000003 011504
006032 004567 173226
006036 016225
006040 016767 011472 173514
006046 004767 173304
006052 004767 005644
006056 004567 005566
006062 005067 011410
006066 005067 011406
006072 005000
006074 012701 000001
006100 012767 017556 011400
006106 012767 000400 011360
006114 005002
006116 010162 017556
006122 005722
006124 022702 000400
006130 001372
006132 004567 004362
006136 000003
006140 105777 011274
006144 100375
006146 005777 011266
006152 100002
006154 104400
006156 000475
006160 004567 004334
006164 000007
006166 105777 011246
006172 100375
006174 005777 011240
006200 100012
006202 104400
006204 004567 173054
006210 016345
006212 016767 011340 173342
006220 004767 173120
006224 000452
006226 005700
006230 001411
006232 005002
006234 012762 177777 017556 215:
006242 005722
006244 022702 000400
006250 001371

.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 R5,PRINTS ;PRINT MESSAGE
MES6
MOV TESTNO,TTY
JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
RWRCK: JSR PC,INIT ;INITIALIZE
JSR R5,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 R5,FUNCT ;WRITE PATTERN
.WORD 3
TSTB @RPCS ;WAIT FOR DONE
BPL -4
TST @RPCS ;ANY DEVICE ERRORS?
BPL 2\$;BRANCH IF NO
HLT ;ERROR AFTER WRITING ONE SECTOR
BR 3\$
2\$: JSR R5,FUNCT ;WRITE CHECK THE DATA
.WORD 7
TSTB @RPCS ;WAIT FOR DONE
BPL -4
TST @RPCS ;ANY DEVICE ERRORS?
BPL 4\$;BRANCH IF NO
HLT ;ERROR AFTER WRITE CHECK OPERATION
JSR R5,PRINTS ;PRINT MESSAGE
MES10
MOV OUTBUF,TTY
JSR PC,PRINTR ;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
BNE 21\$


```

1644 006252 000407          BR      22$
1645 006254 005002          20$: CLR      R2
1646 006256 005062 017556 5$: CLR      OUTBUF(R2) ;CLEAR OUTPUT BUFFER
1647 006262 005722          TST      (R2)+
1648 006264 022702 000400          CMP      #400,R2 ;ENTIRE BUFFER CLEAR?
1649 006270 001372          BNE      5$ ;BRANCH IF NO
1650 006272 004567 004222          22$: JSR      R5,FUNCT ;WRITE CHECK THE DATA AND
1651 006276 000007          .WORD   7 ;EXPECT AN ERROR
1652 006300 105777 011134          TSTB    @RPCS ;WAIT FOR DONE
1653 006304 100375          BPL      -4
1654 006306 032777 000010 011142          BIT      @B3,@RPER ;IS WRITE CHECK ERROR SET?
1655 006314 001011          BNE      6$ ;BRANCH IF YES
1656 006316 104400          HLT
1657 006320 004567 172740          JSR      R5,PRINT$ ;WRITE CHECK ERROR DID NOT SET
1658 006324 016345          MES10 ;PRINT MESSAGE
1659 006326 010167 173230          MOV      R1,TTY
1660 006332 004767 173006          JSR      PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
1661 006336 000405          BR      3$
1662 006340 005777 0111074 6$: TST      @RPCS ;DID ERROR FLAG SET?
1663 006344 100402          BMI     3$ ;BRANCH IF YES
1664 006346 104400          HLT ;ERROR FLAG DID NOT SET AFTER WRITE CHECK ERROR
1665 006350 000400          BR      3$
1666 006352 012767 006114 172422 3$: MOV      #23$,LAD ;SETUP LOOP ADDR
1667 006360 104000          SCOPE
1668 006362 005700          TST      R0 ;ARE WE FLOATING A ONE?
1669 006364 001013          BNE      9$ ;BRANCH IF NO
1670 006366 000241          CLC
1671 006370 006101          ROL      R1 ;ROTATE PATTERN
1672 006372 103402          BCS     10$ ;BRANCH IF COMPLETE
1673 006374 000167 177514          JMP      23$
1674 006400 012700 000001          10$: MOV      #1,R0 ;SET PATTERN FLAG
1675 006404 012701 077777          MOV      #077777,R1 ;SET NEW PATTERN IN R1
1676 006410 000167 177500          JMP      23$
1677 006414 000241          9$: CLC
1678 006416 006201          ASR      R1 ;ROTATE FLOATING ZERO PATTERN
1679 006420 052701 100000          BIS      @B15,R1
1680 006424 103002          BCC     PATFIL ;HAS ZERO BEEN FLOATED
1681 006426 000167 177462          JMP      23$ ;JUMP IF NO
1682
1683
1684
1685          ;CHECK THE ABILITY OF THE RP11C TO CLEAR THE REMAINDER OF A SECTOR
1686          ;ON A PARTIAL WRITE OPERATION. A SECTOR OF ALL ONES IS WRITTEN AND
1687          ;THEN A TWO WORD WRITE OPERATION IS PERFORMED. THE SECTOR IS THEN
1688          ;READ BACK AND VERIFIED. THE FIRST TWO WORDS SHOULD BE ONES AND
1689          ;THE REST SHOULD BE ZEROS.
1690
1691 006432 012701 017556          PATFIL: MOV      #OUTBUF,R1
1692 006436 012700 177777          MOV      #177777,R0
1693 006442 012702 000400          MOV      #400,R2
1694 006446 010021          1$: MOV      R0,(R1)+ ;GENERATE ALL ONES PATTERN
1695 006450 005302          DEC      R2
1696 006452 001375          BNE      1$
1697 006454 004567 004040          JSR      R5,FUNCT ;WRITE SECTOR WITH ONES
1698 006460 000003          .WORD   3
1699 006462 105777 010752          TSTB    @RPCS ;WAIT FOR DONE
  
```

```

1700 006466 100375          BPL      -4
1701 006470 005777 010744  TST      2RPCS      ;ANY DEVICE ERRORS
1702 006474 100002          BPL      2$          ;BRANCH IF NO
1703 006476 104400          HLT                               ;ERROR AFTER WRITING ONE SECTOR ALL 1'S
1704 006500 000473          BR                               ;
1705 006502 012767 000002 010764 2$: MOV      2,WRDCT      ;SETUP FOR TWO WORD WRITE
1706 006510 004567 004004          JSR      R5,FUNCT    ;WRITE TWO WORD
1707 006514 000003          .WORD 3
1708 006516 105777 010716  TSTB     2RPCS      ;WAIT FOR DONE
1709 006522 100375          BPL      -4
1710 006524 005777 010710  TST      2RPCS      ;ANY ERRORS?
1711 006530 100002          BPL      4$          ;BRANCH IF NO
1712 006532 104400          HLT                               ;ERROR ON ONE WORD WRITE
1713 006534 000455          BR                               ;
1714 006536 012767 000400 010730 4$: MOV      400,WRDCT   ;SETUP WORD COUNT
1715 006544 004567 003750          JSR      R5,FUNCT    ;READ SECTOR
1716 006550 000005          .WORD 5
1717 006552 105777 010662  TSTB     2RPCS      ;WAIT FOR DONE
1718 006556 100375          BPL      -4
1719 006560 005777 010654  TST      2RPCS      ;ANY ERRORS
1720 006564 100006          BPL      5$          ;BRANCH IF NO
1721 006566 104400          HLT                               ;ERROR AFTER READING ONE SECTOR
1722 006570 032777 040000 010642  BIT      814,2RPCS   ;WAS IT A DATA ERROR?
1723 006576 001401          BEQ      5$          ;BRANCH IF YES
1724 006600 000433          BR                               ;
1725 006602 022767 177777 010746 5$: CMP      177777,OUTBUF ;COMPARE FIRST WORD SHOULD BE ONES
1726 006610 001410          BEQ      6$          ;BRANCH IF OK
1727 006612 012767 177777 005732  MOV      177777,EXPS
1728 006620 016767 010732 005726  MOV      OUTBUF,RECS
1729 006626 104401          HLT      +1          ;DATA COMPARE ERROR ON FIRST
1730 006630 000417          BR                               ;WORD READ
1731 006632 012700 017562          6$: MOV      OUTBUF+4,R0
1732 006636 012701 000374          MOV      374,R1
1733 006642 005720          8$: TST      (R0)+
1734 006644 001003          BNE      7$          ;REMAINDER OF SECTOR SHOULD BE CLEAR
1735 006646 005301          DEC      R1          ;BRANCH IF NO
1736 006650 001374          BNE      8$
1737 006652 000406          BR      3$
1738 006654 016067 177776 005672 7$: MOV      -2(R0),RECS
1739 006662 005067 005664          CLR      EXPS
1740 006666 104401          HLT      +1          ;DATA FOUND IN AREA OF SECTOR
1741                                ;WHICH SHOULD HAVE BEEN CLEARED
1742                                ;BY A ONE WORD WRITE
1743 006670 012767 006432 172104 3$: MOV      PATFIL,LAD ;SET UP LOOP ADDR
1744 006676 104000          SCOPE
1745
1746                                ;CHECK THE SETTING OF EOP WHEN TRYING TO WRITE BEYOND
1747                                ;THE LIMITS OF THE PACK. THE FIRST SECTOR OF THE PACK IS
1748                                ;WRITTEN WITH ZEROS. THEN A TWO SECTOR WRITE OF ALL
1749                                ;ONE'S IS ISSUED FOR CYLINDER 312, HEAD 23, AND SECTOR 11.
1750                                ;EOP AND ERROR BITS SHOULD SET. THE FIRST SECTOR OF THE
1751                                ;PACK IS CHECKED TO MAKE SURE IT IS STILL ZERO.
1752
1753 00700 005067 010574          EOPTST: CLR      DMA          ;CLEAR DISK ADDRESS
1754 00704 005067 010566          CLR      CYLINDER
1755 006710 012767 000400 010556  MOV      400,WRDCT    ;SET WORDCOUNT TO ONE SECTOR

```

1756	006716	012767	017556	010562		MOV	#OUTBUF, BUF	; SETUP OUTPUT BUFFER
1757	006724	005001				CLR	R1	
1758	006726	005061	017556		1\$:	CLR	OUTBUF(R1)	; CLEAR THE OUTPUT BUFFER
1759	006732	005721				TST	(R1)+	
1760	006734	022701	000400			CMP	#400, R1	
1761	006740	001372				BNE	1\$	
1762	006742	004567	003552			JSR	R5, FUNCT	; WRITE SECTOR ZERO WITH ZEROS
1763	006746	000003			.WORD	3		
1764	006750	105777	010464			TSTB	DRPCS	; WAIT FOR DONE
1765	006754	100375				BPL	-4	
1766	006756	005777	010456			TST	DRPCS	; ANY DEVICE ERRORS
1767	006762	100002				BPL	2\$; BRANCH IF NO
1768	006764	104400				HLT		; ERROR AFTER WRITING SECTOR ZERO WITH ZEROS
1769	006766	000502				BR	3\$	
1770	006770	012767	001000	010476	2\$:	MOV	#1000, WRDCT	; SET WORDCOUNT EQUAL TO TWO SECTORS
1771	006776	112767	000312	010472		MOVB	#312, CYLINDER	; SELECT CYLINDER 312
1772	007004	012767	000011	010466		MOV	#11, DMA	; SELECT RECTOR 11
1773	007012	112767	000023	010461		MOVB	#23, DMA+1	; SELECT HEAD 23
1774	007020	012702	177777			MOV	#177777, R2	
1775	007024	005001				CLR	R1	
1776	007026	010261	017556		4\$:	MOV	R2, OUTBUF(R1)	; SET OUTPUT BUFFER TO ONES
1777	007032	005721				TST	(R1)+	
1778	007034	022701	001000			CMP	#1000, R1	
1779	007040	001372				BNE	4\$	
1780	007042	004567	003452			JSR	R5, FUNCT	; ISSUE TWO SECTOR WRITE TO
1781	007046	000003			.WORD	3		; CYLINDER 312, HEAD 23, AND SECTOR 11
1782	007050	105777	010364			TSTB	DRPCS	; WAIT FOR DONE
1783	007054	100375				BPL	-4	
1784	007056	032777	000002	010372		BIT	#B1, DRPER	; DID EOP ERROR FLAG SET?
1785	007064	001002				BNE	5\$; BRANCH IF SET
1786	007066	104400				HLT		; EOP ERROR FLAG DID NOT SET WHEN
1787	007070	000441				BR	3\$; WRITE OPERATOR EXCEEDS THE PACK
1788	007072	032777	100000	010340	5\$:	BIT	#B15, DRPCS	; DID THE ERROR FLAG SET?
1789	007100	001002				BNE	6\$; BRANCH IF SET
1790	007102	104400				HLT		; ERROR DID NOT SET AFTER GENERATING
1791	007104	000433				BR	3\$; EOP
1792	007106	012767	000002	010360	6\$:	MOV	#2, WRDCT	
1793	007114	005067	010360			CLR	DMA	; CLEAR THE DISK ADDRESS
1794	007120	005067	010352			CLR	CYLINDER	
1795	007124	004567	003370			JSR	R5, FUNCT	; READ THE FIRST SECTOR OF THE PACK
1796	007130	000005			.WORD	5		; AND EXPECT TO FIND ZEROS
1797	007132	105777	010302			TSTB	DRPCS	; WAIT FOR READY
1798	007136	100375				BPL	-4	
1799	007140	005777	010274			TST	DRPCS	; WERE THERE ANY ERRORS?
1800	007144	100002				BPL	7\$; BRANCH IF NO
1801	007146	104400				HLT		; ERROR ENCOUNTERED ON 2 WORD READ
1802	007150	000411				BR	3\$; OF FIRST SECTOR ON THE PACK
1803	007152	016767	010400	005374	7\$:	MOV	OUTBUF, RECS	; GET FIRST WORD OF BUFFER
1804	007160	005767	005370			TST	RECS	; DOES 1ST SECTOR STILL CONTAIN ZEROS?
1805	007164	001403				BEQ	3\$; BRANCH IF YES
1806	007166	005067	005360			CLR	EXPS	
1807	007172	104401				HLT	+1	; CONTENTS OF THE FIRST SECTOR OF THE
1808								; PACK CHANGED AFTER FORCING EOP
1809								; ERROR, OPERATION PROBABLY
1810								; WRAPPED AROUND.
1811	007174	012767	006700	171600	3\$:	MOV	#EOPTST, LAD	

E04

RP11C RELIABILITY TEST MACY11 27(732) 16-SEP-76 16:12 PAGE 43
DZRPF8.P11 ***** TEST 3 *****

1812	007202	104000			SCOPE		
1813	007204	032737	002000	177570	BIT	#B10,3#SWR	:LOOP ON TEST?
1814	007212	001402			BEQ	MEMTST	:BRANCH IF NO
1815	007214	000167	176632		JMP	RWRCK	

1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835
 1836
 1837
 1838
 1839
 1840
 1841
 1842
 1843
 1844
 1845
 1846
 1847
 1848
 1849
 1850
 1851
 1852
 1853
 1854
 1855
 1856
 1857
 1858
 1859
 1860
 1861
 1862
 1863
 1864
 1865
 1866
 1867
 1868
 1869
 1870
 1871

007220 012767 000004 010310
 007226 004567 172032
 007232 016225
 007234 016767 010276 172320
 007242 004767 172110
 007246 012767 177700 010250
 007254 004767 004442
 007260 016700 006416
 007264 162700 017556
 007270 000241
 007272 006000
 007274 042700 000001
 007300 010067 010170
 007304 012702 017556
 007310 012703 017556
 007314 010322
 007316 005723
 007320 020267 006356
 007324 101773
 007326 012767 017556 010152
 007334 005067 010140
 007340 005067 010132
 007344 004567 003150
 007350 000003
 007352 105777 010062
 007356 100375
 007360 005777 010054
 007364 100002
 007366 104400
 007370 000446
 007372 016700 010076
 007376 012701 017556
 007402 005021
 007404 005300
 007406 001375
 007410 004567 003104
 007414 000005
 007416 105777 010016
 007422 100375
 007424 005777 010010
 007430 100006
 007432 104400
 007434 032777 040000 007776
 007442 001401
 007444 000420

.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.

```
MEMTST: MOV #4,TESTNO
          JSR R5,PRINTS ;PRINT MESSAGE
          MES6
          MOV TESTNO,TTY
          JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
          MOV #-100,PASSC ;SETUP ITERATION COUNT
RMENT: JSR PC,INIT ;INITIALIZE
        MOV MEMSIZ,RO ;GET TOP OF CORE
        SUB #OUTBUF,RO ;DETERMINE SIZE OF BUFFER IN BYTES
        CLC
        ROR RO ;CONVERT TO WORDS
        BIC #1,RO ;KEEP NUMBER EVEN
        MOV RO,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
2$: BR 3$
    MOV WRDCT,RO
    MOV #OUTBUF,R1
10$: CLR (R1)+ ;CLEAR THE BUFFER
    DEC RO
    BNE 10$
    JSR R5,FUNCT ;READ ADDRESS PATTERN
.WORD 5
    TSTB @RPCS ;WAIT FOR DONE
    BPL -4
    TST @RPCS ;ANY ERRORS?
    BPL 4$ ;BRANCH IF NO
    HLT ;ERROR AFTER READING ADDR PATTERN
    BIT #B14,@RPCS ;IS THIS A DATA ERROR?
    BEQ 4$ ;BRANCH IF YES
    BR 3$
```

```

1872 007446 012702 017556 4$: MOV #OUTBUF,R2
1873 007452 010203 MOV R2,R3 ;COMPARE THE ADDR PATTERN
1874 007454 020322 6$: CMP R3,(R2)+ ;IS DATA CORRECT?
1875 007456 001005 BNE 5$ ;BRANCH IF NO
1876 007460 005723 TST (R3)+
1877 007462 020267 006214 CMP R2,MEMSIZ ;IS ENTIRE BUFFER VERIFIED?
1878 007466 101772 BLOS 6$ ;BRANCH IF NO
1879 007470 000406 BR 3$
1880 007472 010367 005054 5$: MOV R3,EXPS
1881 007476 016267 177776 005050 MOV -2(R2),RECS
1882 007504 104401 HLT +1 ;COMPARE ERROR UTILIZING ADDR PATTERN
1883 007506 012767 007304 171266 3$: MOV #7$,LAD ;SETUP LOOP ADD
1884 007514 104000 SCOPE
1885
1886 ;CHECK THE SETTING OF NON EXISTENT MEMORY FLAG IN RPER.
1887
1888 007516 012767 000001 007750 NONEX: MOV #1,WRDCT ;SETUP WORD COUNT
1889 007524 012767 160000 007754 MOV #16000,BUF ;LOAD ADDR TO FORCE NON EX MEMORY
1890 007532 004567 002762 JSR R5,FUNCT ;ISSUE A WRITE COMMAND AND
1891 007536 000063 .WORD 63 ;EXPECT NON EX MEMORY
1892 007540 105777 007674 TSTB JRPCS ;WAIT FOR READY
1893 007544 100375 BPL -4
1894 007546 032777 000004 007702 BIT #B2,JRPER ;DID NON EX MEMORY SET
1895 007554 001002 BNE 1$ ;BRANCH IF SET
1896 007556 104400 HLT ;NON EX MEMORY DID NOT SET
1897 007560 000413 BR 2$
1898 007562 032777 040000 007650 1$: BIT #B14,JRPCS ;DID HARD ERROR SET ON NON EX MEMORY
1899 007570 001002 BNE 3$ ;BRANCH IF SET
1900 007572 104400 HLT ;HARD ERROR DID NOT SET
1901 007574 000405 BR 2$
1902 007576 032777 100000 007634 3$: BIT #B15,JRPCS ;DID THE ERROR FLAG SET AFTER NON EX MEMORY
1903 007604 001001 BNE 2$ ;BRANCH IF SET
1904 007606 104400 HLT ;ERROR FLAG DID NOT SET
1905 007610 012767 007516 171164 2$: MOV #NONEX,LAD
1906 007616 104000 SCOPE
1907
1908 ;IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS
1909 ;BITS.
1910
1911 007620 012767 010312 170156 EXTTST: MOV #EXTTRP,4 ;SETUP TIMEOUT TRAP
1912 007626 012767 000340 170152 MOV #PRI7,6
1913 007634 005737 177572 TST J#SRO ;IF MEMORY MANAGEMENT IS NOT
1914 ;AVAILABLE THE PROGRAM WILL TRAP
1915 ;AND TRANSFER TO END OF THE TEST
1916 007640 012767 010306 170136 MOV #EXTRP,4
1917 007646 012737 007600 172356 MOV #7600,J#KIPAR7 ;OPEN I/O REGISTERS
1918 007654 005037 172340 CLR J#KIPAR0 ;FREE FIRST 4K
1919 007660 012737 002000 172342 MOV #2000,J#KIPAR1
1920 007666 012737 177406 172300 MOV #400*256.-400+UP+RW,J#KIPDR0 ;SET KIPDR0=RW UP 400 BLOCKS
1921 007674 012737 177406 172302 MOV #400*256.-400+UP+RW,J#KIPDR1 ;SET KIPDR1=RW UP 400 BLOCKS
1922 007702 012737 177406 172316 MOV #400*256.-400+UP+RW,J#KIPDR7 ;SET KIPDR7=RW UP 400 BLOCKS
1923 007710 012737 000001 177572 MOV #1,J#SRO ;TURN ON MEMORY MANAGEMENT
1924 007716 012702 020000 MOV #20000,R2 ;R2 EQUALS BASE ADDR
1925 007722 012712 177777 7$: MOV #177777(R2) ;INSERT PATTERN INTO 200000
1926 007726 012767 000002 007540 MOV #2,WRDCT ;SETUP WORDCOUNT
1927 007734 005067 007546 CLR BUF ;CLEAR BUFFER ADDR

```

1928	007740	004567	002554		JSR	R5,FUNCT				
1929	007744	000023		.WORD	23					;WRITE TWO WORDS ON DISK STARTING ;AT LOCATION 200000
1930										;TO SET MEXO
1931	007746	105777	007466		TSTB	DRPCS				;WAIT FOR READY
1932	007752	100375			BPL	-4				
1933	007754	005777	007460		TST	DRPCS				
1934	007760	100002			BPL	1\$				
1935	007762	104400			HLT					;STATUS ERROR AFTER 2 WORD WRITE
1936	007764	000441			BR	2\$;USING MEXO
1937	007766	032777	000020	007444	1\$:	BIT	#B4,DRPCS			;MEXO SHOULD HAVE SET?
1938	007774	001002			BNE	3\$;BRANCH IF SET
1939	007776	104400			HLT					;MEXO DID NOT SET
1940	010000	000433			BR	2\$				
1941	010002	005012		3\$:	CLR	(R2)				;CLEAR LOCATION 200000
1942	010004	004567	002510		JSR	R5,FUNCT				;READ TWO WORDS INTO LOCATIONS ;200000 AND 200001.
1943	010010	000025		.WORD	25					;WAIT FOR READY
1944	010012	105777	007422		TSTB	DRPCS				
1945	010016	100375			BPL	-4				
1946	010020	005777	007414		TST	DRPCS				;ANY ERRORS?
1947	010024	100002			BPL	4\$;BRANCH IF NO
1948	010026	104400			HLT					;ERROR OFTER READING 2 WORDS
1949	010030	000417			BR	2\$				
1950	010032	032777	000020	007400	4\$:	BIT	#B4,DRPCS			;DID MEXO SET?
1951	010040	001002			BNE	5\$;BRANCH IF YES
1952	010042	104400			HLT					;MEXO DID NOT SET AFTER 2 WORD
1953	010044	000411			BR	2\$;READ STARTING AT 200000
1954	010046	022712	177777	5\$:	CMP	#177777,(R2)				;WAS DATA READ INTO LOCATION ;200000 CORRECTLY? - BRANCH IF YES
1955	010052	001406			BEG	2\$				
1956	010054	012767	177777	004470	MOV	#177777,EXPS				
1957	010062	011267	004466		MOV	(R2),RECS				
1958	010066	104401			HLT	+1				;DATA COMPARE ERROR AT 200000
1959	010070	012767	007722	170704	2\$:	MOV	#7\$,LAD			;SETUP ERROR LOOP
1960	010076	104000			SCOPE					
1961	010100	012737	004000	172342	EXTT1:	MOV	#4000,DRKIPARI			
1962	010106	012702	020000		MOV	#20000,R2				;R2 EQUALS THE BASE ADDR
1963	010112	012712	177777	7\$:	MOV	#177777,(R2)				;INSERT PATTERN INTO 400000
1964	010116	004567	002376		JSR	R5,FUNCT				
1965	010122	000043		.WORD	43					
1966	010124	105777	007310		TSTB	DRPCS				;WAIT FOR READY
1967	010130	100375			BPL	-4				
1968	010132	005777	007302		TST	DRPCS				;ANY ERRORS?
1969	010136	100002			BPL	4\$;BRANCH IF NO
1970	010140	104400			HLT					
1971	010142	000441			BR	2\$;ERROR AFTER READING 2 WORDS
1972	010144	032777	000040	007266	4\$:	BIT	#B5,DRPCS			;DID MEXI SET?
1973	010152	001002			BNE	10\$;BRANCH IF YES
1974	010154	104400			HLT					;MEXI DI NOT SET WITH A TWO WORD
1975										;TRANSFER STARTING AT 400000
1976	010156	000433			BR	2\$				
1977	010160	005012		10\$:	CLR	(R2)				;CLEAR LOCATION 400000
1978	010162	004567	002332		JSR	R5,FUNCT				;READ TWO WORDS STARTING AT 400000
1979	010166	000045		.WORD	45					
1980	010170	105777	007244		TSTB	DRPCS				;WAIT FOR READY
1981	010174	100375			BPL	-4				
1982	010176	005777	007236		TST	DRPCS				;ANY ERRORS?
1983	010202	100002			BPL	11\$;BRANCH IF NO

1984	010204	104400				HLT			;ERROR WHILE READING TWO WORDS
1985	010206	000417				BR	2\$		
1986	010210	022712	177777		11\$:	CMP	#177777,(R2)		;WAS DATA READ INTO LOCATION 400000
1987	010214	001407				BEQ	6\$;CORRECTLY? - BRANCH IF YES
1988	010216	012767	177777	004326		MOV	#177777,EXPS		
1989	010224	011267	004324			MOV	(R2),RECS		
1990	010230	104401				HLT	+1		;DATA COMPARE ERROR AT 400000 IF
1991	010232	000405				BR	2\$;RECEIVED=0 - LOCATION WASN'T ACCESSED
1992	010234	032777	000040	007176	6\$:	BIT	#B5,2RPCS		;DID MEX1 SET?
1993	010242	001001				BNE	2\$		
1994	010244	104400				HLT			;MEX1 DID NOT SET AFTER 2 WORD TRANSFER
1995									;STARTING AT 377777
1996	010246	012767	010112	170526	2\$:	MOV	#7\$,LAD		;SETUP ERROR LOOP
1997	010254	104000				SCOPE			
1998	010256	000413				BR	EXTRP		
1999	010260	005267	007240		EXTEND:	INC	PASSC		;INCREMENT ITERATION COUNT
2000	010264	001402				BEQ	1\$		
2001	010266	000167	176762			JMP	RMEMT		
2002	010272	032737	002000	177570	1\$:	BIT	#B10,2\$SWR		;LOOP ON TEST?
2003	010300	001414				BEQ	DATA		
2004	010302	000167	176746			JMP	RMEMT		
2005									
2006	010306	005037	177572		EXTRP:	CLR	2\$SR0		;TURN OFF MEMORY MANAGEMENT
2007	010312	012706	000500		EXTTRP:	MOV	#STKPTR,SP		;RESTORE STACK
2008	010316	012767	000006	167460		MOV	#6,4		
2009	010324	005067	167456			CLR	6		
2010	010330	000753				BR	EXTEND		

2011
2012
2013
2014
2015
2016
2017
2018
2019
2020 010332 012767 000005 007176
2021 010340 004567 170720
2022 010344 016225
2023 010346 016767 007164 171206
2024 010354 004767 170776
2025 010360 016700 007120
2026 010364 000241
2027 010366 006000
2028 010370 010037 177570
2029 010374 005067 007076
2030 010400 005067 007074
2031 010404 012737 000200 177776
2032 010412 016767 007072 007054
2033 010420 012737 012602 000254
2034 010426 012737 000340 000256
2035 010434 004767 003460
2036 010440 012767 017556 007040
2037 010446 032767 040000 007010
2038 010454 001424
2039 010456 004767 002520
2040 010462 004567 002032
2041 010466 000103
2042 010470 032737 010000 177570
2043 010476 001003
2044 010500 004767 005200
2045 010504 000401
2046 010506 000001
2047 010510 012767 010456 170264
2048 010516 104000
2049 010520 004767 002514
2050 010524 000754
2051 010526 032767 020000 006730
2052 010534 001424
2053 010536 004767 002440
2054 010542 004567 001752
2055 010546 000107
2056 010550 032737 010000 177570
2057 010556 001003
2058 010560 004767 005120
2059 010564 000401
2060 010566 000001
2061 010570 012767 010536 170204
2062 010576 104000
2063 010600 004767 002434
2064 010604 000754
2065 010606 032767 010000 006650
2066 010614 001530

.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.

DATAT: MOV #5,TESTNO
JSR R5,PRINTS ;PRINT MESSAGE
MES6
MOV TESTNO,TTY
JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
RDATAT: MOV PATNU,R0 ;GET PATTERN NO.
CLC
ROR R0
MOV R0,#SWR ;DISPLAY PATTERN NO. IN USE
CLR CYLINDER
CLR DMA
MOV #PRI4,#PSW ;ENABLE INT SYSTEM
MOV SWRDCT,WRDCT
MOV #DKINT,#VECTOR ;SETUP DISK VECOTR
MOV #340,#STATUS
DATP: JSR PC,PASEL ;GENERATE PATTERN
MOV #OUTBUF,BUF ;SETUP BUFFER ADDR
BIT #B14,FLAG ;WRITE?
BEQ WRICK ;BRANCH IF NO
LDAT: JSR PC,OPDSEL ;ANY OPERATOR ADDR PARAMETERS?
JSR R5,FUNCT ;WRITE WITH INTERRUPTS
.WORD 103
BIT #B12,#SWR ;DETERMINE HOW TO WAIT FOR INT
BNE 1\$
JSR PC,NPR ;GENERATE WORSE CASE NPR CYCLES
BR 2\$
1\$: WAIT
2\$: MOV #LDAT,LAD ;SETUP LOOP ADDR
SCOPE
JSR PC,DISBUF ;PREPARE NEW DISK ADDR
BR LDAT
WRICK: BIT #B13,FLAG ;WRITE CHECK?
BEQ DREAD ;BRANCH IF NO
3\$: JSR PC,OPDSEL ;ANY OPERATOR ADDR PARAMETERS?
JSR R5,FUNCT ;WRITE CHECK THE DATA
.WORD 107
BIT #B12,#SWR ;DETERMINE HOW TO WAIT FOR INT
BNE 1\$
JSR PC,NPR ;GENERATE WORSE CASE NPR CYCLES
BR 2\$
1\$: WAIT
2\$: MOV #3\$,LAD ;SETUP LOOP ADDR
SCOPE
JSR PC,DISBUF ;PREPARE NEW DISK ADDR
BR 3\$
DREAD: BIT #B12,FLAG ;READ?
BEQ MSTR ;BRANCH IF NO

2067	010616	005067	006706	ESH:	CLR	RDERR	;CLEAR READ ERROR COUNT
2068	010622	004767	002354		JSR	PC,OPDSEL	;ANY OPERATOR ADDR PARAMETERS?
2069	010626	005067	173456	DSKRD:	CLR	INTFLG	;CLEAR THE INTERRUPT FLAG
2070	010632	004567	001662		JSR	RS,FUNCT	;READ THE DATA
2071	010636	000105		.WORD	105		
2072	010640	032737	001000 177570		BIT	#B9,@#SWR	;COMPARE DATA?
2073	010646	001411			BEQ	1\$;BRANCH IF YES
2074	010650	032737	010000 177570		BIT	#B12,@#SWR	;DETERMINE HOW TO WAIT FOR INT
2075	010656	001003			BNE	2\$	
2076	010660	004767	005020		JSR	PC,NPR	;GENERATE WORSE CASE NPR CYCLES
2077	010664	000431			BR	3\$	
2078	010666	000001		2\$:	WAIT		
2079	010670	000427			BR	3\$	
2080	010672	016700	006576	1\$:	MOV	WRDCT,RO	;CLEAR THE INPUT BUFFER
2081	010676	012702	000012		MOV	#12,R2	
2082	010702	012701	017556		MOV	#OUTBUF,R1	
2083	010706	005021		4\$:	CLR	(R1)+	
2084	010710	005021			CLR	(R1)+	
2085	010712	005021			CLR	(R1)+	
2086	010714	005021			CLR	(R1)+	
2087	010716	005021			CLR	(R1)+	
2088	010720	005021			CLR	(R1)+	
2089	010722	005021			CLR	(R1)+	
2090	010724	005021			CLR	(R1)+	
2091	010726	005021			CLR	(R1)+	
2092	010730	005021			CLR	(R1)+	
2093	010732	160200			SUB	R2,RO	
2094	010734	100364			BPL	4\$	
2095	010736	004767	003650		JSR	PC,COMPAR	;COMPARE THE DATA
2096	010742	105777	006472		TSTB	@RPCS	;WAIT FOR READY
2097	010746	100375			BPL	-4	
2098	010750	005767	006560	3\$:	TST	INTERR	;WERE THERE ANY ERRORS
2099	010754	001424			BEQ	5\$;BRANCH IF NO
2100	010756	005267	006546		INC	RDERR	;UPDATE ERROR COUNT
2101	010762	022767	000024 006540		CMP	#20.,RDERR	;MORE THAN 20 ERRORS?
2102	010770	001416			BEQ	5\$;BRANCH IF YES
2103	010772	022767	000012 006530		CMP	#10.,RDERR	;IS THIS TENTH ERROR?
2104	011000	001312			BNE	DSKRD	;BRANCH IF NO
2105	011002	112777	000015 006430		MOVB	#15,@RPCS	;HOME THE HEADS
2106	011010	105777	006424		TSTB	@RPCS.	;WAIT FOR DONE
2107	011014	100375			BPL	-4	
2108	011016	005777	006436		TST	@RPDS	
2109	011022	100375			BPL	-4	;WAIT FOR READY
2110	011024	000700			BR	DSKRD	
2111	011026	005767	006476	5\$:	TST	RDERR	
2112	011032	001410			BEQ	6\$	
2113	011034	004567	170224		JSR	RS,PRINTS	;PRINT MESSAGE
2114	011040	016240			MES7		
2115	011042	016767	006462 170512		MOV	RDERR,TTY	
2116	011050	004767	170302		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2117	011054	005067	006450	6\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
2118	011060	012767	010616 167714		MOV	#ESH,LAD	;LOOP ADDR
2119	011066	104000			SCOPE		
2120	011070	004767	002144		JSR	PC,DISBUF	;GET NEW DISK ADDR
2121	011074	000650			BR	ESH	
2122	011076	032767	000040 006360	MSTR:	BIT	#B5,FLAG	;LOOPING ON AN OPERATOR ADDR?

2123	011104	001402			BEQ	1\$:NO CONTINUE
2124	011106	000167	177322		JMP	DATP	:YES
2125	011112	005767	006346	1\$:	TST	FLAG	
2126	011116	100002			BPL	2\$:UNDER PROGRAM CONTROL
2127	011120	000167	000614		JMP	MULCHK	:OPERATOR SELECTED PATTERN
2128	011124	062767	000002	006352	2\$:	ADD	:INC PATTERN INDEX
2129	011132	022767	000036	006344		CMP	:PATTERNS EXCEEDED?
2130	011140	001402			BEQ	3\$	
2131	011142	000167	177212		JMP	RDATAT	:NOT YET
2132	011146	005067	006332		3\$:	CLR	:LAST PATTERN USED
2133	011152	032737	002000	177570		BIT	:LOOP ON TEST?
2134	011160	001402			BEQ	RANEX	:NO..GO TO RANDOM TEST
2135	011162	000167	177172		JMP	RDATAT	:YES

```

2136 .SBTTL ***** TEST 6 *****
2137
2138 ;THIS IS A RANDOM ADDRESS AND DATA TEST.
2139
2140 011166 012767 000006 006342 RANEX: MOV #6,TESTNO
2141 011174 004567 170064 JSR R5,PRINTS ;PRINT MESSAGE
2142 011200 016225 MES6
2143 011202 016767 006330 170352 MOV TESTNO,TTY
2144 011210 004767 170142 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2145 011214 004767 002502 RRANEX: JSR PC,INIT
2146 011220 012767 000034 006256 MOV #34,PATNU
2147 011226 012767 173000 006270 MOV #-5000,PASSC ;SET UP PASS COUNT
2148 011234 012737 000200 177776 MOV #PRI4,#PSW
2149 011242 012767 000400 006224 WRLG: MOV #400,WRDCT ;SET UP WORD COUNT TO 1 SECTOR
2150 011250 016767 006220 006264 MOV WRDCT,WORK
2151 011256 012701 017556 MOV #OUTBUF,R1
2152 011262 004767 002724 JSR PC,RANDOM ;GENERATE RANDOM PATTERN
2153 011266
2154 011266 004767 170272 1$: JSR PC,RAND$ ;GENERATE TWO RANDOM NOS.
2155 011272 016767 170404 006242 MOV LONUM,WORK
2156 011300 016767 170374 006236 MOV HINUM,WORK1
2157 011306 042767 177400 006226 BIC #177400,WORK
2158 011314 022767 000312 006220 CMP #312,WORK ;FORM RANDOM CYL ADDR
2159 011322 002761 BLT 1$
2160 011324 016767 006212 006144 MOV WORK,CYLINDER ;SAVE IT
2161 011332 042767 160360 006204 BIC #160360,WORK1
2162 011340 122767 000011 006176 CMPB #11,WORK1 ;FORM RANDOM SECTOR ADDR
2163 011346 101003 BHI 2$
2164 011350 042767 000010 006166 BIC #10,WORK1
2165 011356 122767 000023 006161 2$: CMPB #23,WORK1+1
2166 011364 101003 BHI 3$ ;FORM RANDOM HEAD ADDR
2167 011366 142767 000014 006151 BICB #14,WORK1+1
2168 011374 016767 006144 006076 3$: MOV WORK1,DMA ;SAVE DESK ADDR.
2169 011402 012767 017556 006076 RANLOP: MOV #OUTBUF,BUF ;SETUP OUTPUT BUFFER
2170 011410 004567 001104 JSR R5,FUNCT ;WRITE RANDOM DATA AND
2171 011414 000103 .WORD 103 ;ENABLE INTERRUPTS
2172 011416 032737 010000 177570 BIT #B12,#SWR ;DETERMINE HOW TO WAIT FOR INT
2173 011424 001003 BNE 2$
2174 011426 004767 004252 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
2175 011432 000401 BR 4$
2176 011434 000001 2$: WAIT
2177 011436 012767 011402 167336 4$: MOV #RANLOP,LAD ;SETUP LOOP ADDR
2178 011444 104000 SCOPE
2179 011446 004567 001046 7$: JSR R5,FUNCT ;WRITE CHECK THE DATA AND
2180 011452 000107 .WORD 107 ;ENABLE INTERRUPT
2181 011454 032737 010000 177570 BIT #B12,#SWR ;HOW TO WAIT FOR INT?
2182 011462 001003 BNE 1$
2183 011464 004767 004214 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
2184 011470 000401 BR 5$
2185 011472 000001 1$: WAIT
2186 011474 012767 011446 167300 5$: MOV #7$,LAD ;SETUP LOOP ADDR
2187 011502 104000 SCOPE
2188 011504 005067 006020 8$: CLR RDERR ;CLEAR READ ERROR COUNTER
2189 011510 005067 172574 11$: CLR INTFLG
2190 011514 004567 001000 JSR R5,FUNCT ;READ RANDOM DATA AND
2191 011520 000105 .WORD 105 ;ENABLE INTERRUPT
  
```

N04.

Line	Address	Code	Hex	Hex	Op	Reg	Comment
2192	011522	032737	010000	177570	BIT	#B12,2#SWR	;HOW TO WAIT FOR INT?
2193	011530	001003			BNE	3\$	
2194	011532	004767	004146		JSR	PC,NPR	;TEST WORSE CASE NPR CYCLES
2195	011536	000401			BR	6\$	
2196	011540	000001			WAIT		
2197	011542	032737	001000	177570	3\$: BIT	#B9,2#SWR	;COMPARE FOR ERRORS?
2198	011550	001006			6\$: BNE	9\$;BRANCH IF NO
2199	011552	032777	040000	005660	BIT	#B14,2RPCS	;HARD ERROR?
2200	011560	001002			BNE	9\$;BRANCH IF YES
2201	011562	004767	003024		JSR	PC,COMPARE	;COMPARE DATA FOR ERRORS
2202	011566	005767	005742		9\$: TST	INTERR	;READ ERROR?
2203	011572	001424			BEQ	10\$;BRANCH IF NO
2204	011574	005267	005730		INC	RDERR	;UPDATE ERROR COUNT
2205	011600	022767	000024	005722	CMP	#20.,RDERR	;20 ERRORS YET?
2206	011606	001416			BEQ	10\$;BRANCH IF YES
2207	011610	022767	000012	005712	CMP	#10.,RDERR	;IS THIS TENTH ERROR?
2208	011616	001334			BNE	11\$;BRANCH IF NO
2209	011620	112777	000015	005612	MOV	#15,2RPCS	;ISSUE HOME COMMAND
2210	011626	105777	005606		TST	2RPCS	;WAIT FOR DONE
2211	011632	100375			BPL	-4	
2212	011634	005777	005620		TST	2RPDS	;WAIT FOR READY
2213	011640	100375			BPL	-4	
2214	011642	000722			BR	11\$	
2215	011644	005767	005660		10\$: TST	RDERR	
2216	011650	001410			BEQ	21\$	
2217	011652	004567	167406		JSR	R5,PRINT\$;PRINT MESSAGE
2218	011656	016240			MES7		
2219	011660	016767	005644	167674	MOV	RDERR,TTY	
2220	011666	004767	167464		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2221	011672	005067	005632		21\$: CLR	RDERR	;CLEAR READ ERROR COUNTER
2222	011676	012767	011504	167076	MOV	#8\$,LAD	;SET UP LOOP ADDR
2223	011704	104000			SCOPE		
2224	011706	005267	005612		INC	PASSC	;INCREMENT PASS COUNT
2225	011712	001402			BEQ	12\$;BRANCH IF DONE
2226	011714	000167	177322		JMP	WRLG	;CONTINUE
2227	011720	005067	005560		12\$: CLR	PATNU	
2228	011724	032737	002000	177570	BIT	#B10,2#SWR	;LOOP ON TEST?
2229	011732	001402			BEQ	MULCHK	;NO
2230	011734	000167	177254		JMP	RRANEX	;LOOP
2231							
2232							
2233							
2234							
2235							
2236							
2237							
2238	011740	005067	005534		MULCHK: CLR	DMA	
2239	011744	005067	005526		CLR	CYLINDER	;CLEAR ADDRESS REGISTERS
2240	011750	032767	004000	005506	BIT	#B11,FLAG	;ARE WE IN MULTI DISK MODE
2241	011756	001422			BEQ	REPOEN	;REPORT "END"
2242	011760	016767	005500	005554	MOV	FLAG,WORK	;WHAT DISK ARE WE ON
2243	011766	042767	177743	005546	BIC	#177743,WORK	;IF LAST DISK ON SYSTEM
2244	011774	026767	005542	005516	CMP	WORK,DSKNOR	;REPORT END
2245	012002	001004			BNE	INDRVE	
2246	012004	042767	000034	005452	BIC	#34,FLAG	
2247	012012	000404			BR	REPOEN	;REPORT "END" LAST DISK

```

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

```

012014	062767	000004	005442	INDRVE:	ADD	#4, FLAG	: INC. DISK NO.
012022	000422				BR	EXTPP	: EXERCISE DISK
012024	005267	166750		REPOEN:	INC	ICNT	: INCREMENT PASS COUNTER
012030	004567	167230			JSR	R5, PRINTS	: PRINT MESSAGE
012034	016026				MES1		: REPORT END OF PASS
012036	016767	166736	167516		MOV	ICNT, TTY	: TYPE LOCATION-SUPRESS ZEROS
012044	004767	167306			JSR	PC, PRINTS	: GET MONITOR RETURN ADDRESS
012050	013701	000042			MOV	R#42, R1	: BRANCH IF NOT UNDER MONITOR
012054	001405				BEQ	EXTPP	
012056	000005				RESET		
012060	004711			MEXIT:	JSR	PC, (R1)	: EXIT TO THE MONITOR
012062	000240				NOP		
012064	000240				NOP		
012066	000240				NOP		
012070	000167	171260		EXTPP:	JMP	ADTST	: RECYCLE

.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.

2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320

012074 012706 000500
012100 012767 012470 165716
012106 012767 000340 165712
012114 004567 001530
012120 112777 000312 005322
012126 052777 000011 005304
012134 012700 017556
012140 012720 025252
012144 020067 003532
012150 101773
012152 012767 000312 005316
012160 005067 005314
012164 012767 000400 005302
012172 012767 017556 005306
012200 105777 005234
012204 100375
012206 005777 005246
012212 100375
012214 004567 000300
012220 000003
012222 105777 005212
012226 100375
012230 032777 100000 005202
012236 001401
012240 104400
012242 012767 012074 166532
012250 104000
012252 004567 167006
012256 016152
012260 004567 000234
012264 000005
012266 105777 005146
012272 100375
012274 000771

PFTST: MOV #STKPTR, SP
MOV #PFD, 24 ;SET UP POWER FAIL VECTOR
MOV #PRI7, 26 ;LOCKOUT INTERRUPTS
JSR R5, DSKNOS ;SELECT THE UNIT
MOVB #312, JRPCA ;SELECT CYLINDER 312
BIS #11, JRPCS ;ISSUE SEEK COMMAND
MOV #OUTBUF, RO
1\$: MOV #25252, (RO)+ ;FILL MEMORY WITH CHECKERBOARD
CMP RO, MEMSIZ ;PATTERN
BLOS 1\$
MOV #312, CYLINDER
CLR DMA
MOV #400, WRDCT
MOV #OUTBUF, BUF
TSTB JRPCS ;WAIT FOR DONE
BPL -4
TST JRPDS ;WAIT FOR UNIT READY
BPL -4
JSR R5, FUNCT ;WRITE 1 SECTOR OF CHECKERBOARD
.WORD 3
TSTB JRPCS ;WAIT FOR READY
BPL -4
BIT #B15, JRPCS ;ANY ERRORS?
BEQ 2\$
HLT ;DEVICE ERROR ON WRITE
2\$: MOV #PFTST, LAD
SCOPE
JSR R5, PRINTS ;PRINT MESSAGE
MESS ;HAVE POWER TURNED OFF
3\$: JSR R5, FUNCT ;GO INTO A LOOP READING
.WORD 5 ;THE DISK SURFACE
TSTB JRPCS
BPL -4
BR 3\$

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

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

```

2321 012336 016777 005200 005074      MOV    WORK,ARPCS      ;SELECT THE UNIT
2322 012344 005777 005110              TST    ARPDS
2323 012350 100352              BPL    PFT1
2324 012352 105777 005104              TSTB   ARPCA1        ;SELECTED CYLINDER SHOULD BE ZERO
2325 012356 001411              BEQ    1$
2326 012360 005067 002166              CLR    EXPS
2327 012364 005067 002164              CLR    RECS
2328 012370 117767 005066 002156      MOVB   ARPCA1,RECS
2329 012376 104401              HLT    +1             ;HEADS DID NOT RESTORE TO 0
2330 012400 000427              BR     2$             ;ON POWER FAILURE
2331 012402 012700 017556          1$:    MOV    #OUTBUF,RO
2332 012406 022720 025252          4$:    CMP    #25252,(RO)+ ;DID MEMORY RETAIN PATTERN ON PF
2333 012412 001004              BNE    3$             ;BRANCH IF NO
2334 012414 020067 003262              CMP    RO,MEMSI2
2335 012420 001372              BNE    4$
2336 012422 000416              BR     2$
2337 012424 012767 025252 002120      3$:    MOV    #25252,EXPS
2338 012432 016067 177776 002114      MOV    -2(RO),RECS
2339 012440 104403              HLT    +3             ;CONTENTS OF MEMORY CHANGED AFTER PF
2340 012442 004567 166616              JSR    RS,PRINTS     ;PRINT MESSAGE
2341 012446 016127              MES4
2342 012450 010067 167106              MOV    RO,TTY
2343 012454 004767 166664              JSR    PC,PRINTR     ;TYPE LOCATION WITH LEADING ZEROS
2344 012460
2345 012460 004567 166600          2$:    JSR    RS,PRINTS     ;PRINT MESSAGE
2346 012464 017431              END
2347 012466 000000              HALT                ;END OF TEST
2348
2349
2350          ;POWER FAIL TRAP HANDLER
2351 012470 012767 012500 165326      PFD:   MOV    #PFU,24 ;SET UP POWER UP VECTOR
2352 012476 000774              BR     PFD
2353
2354          ;POWER UP TRAP HANDLER
2355
2356 012500 012767 000026 165316      PFU:   MOV    #26,24 ;SETUP TRAP CATCHER
2357 012506 005067 165314              CLR    26
2358 012512 012706 000500              MOV    #STKPTR,SP   ;RESTORE STACK
2359 012516 000667              BR     PFT1         ;VERIFY POWER DOWN SEQUENCE
2360

```



```

2361 .SBTTL *** SUBROUTINES ***
2362
2363 ;THIS ROUTINE OUTPUTS THE FUNCTION FOUND AT
2364 ;THE CALL + 2.
2365
2366 012520 004567 001124 FUNCT: JSR R5,DSKNOS ;SELECT THE UNIT
2367 012524 016777 004750 004720 MOV DMA,DRPDA ;SETUP DISK ADDR REG
2368 012532 116777 004740 004710 MOVB CYLINDER,DRPCA ;SETUP CYLINDER ADDR REG
2369 012540 016777 004742 004700 MOV BUF,DRPBA ;SETUP BUS ADDR REG
2370 012546 016777 004722 004670 MOV WRDCT,DRPWC ;SETUP WORD COUNT
2371 012554 005477 004664 NEG DRPWC ;COMPLIMENT WORD COUNT
2372 012560 011567 000014 MOV (R5),FNCT ;GET RPCS FUNCTION
2373 012564 062705 000002 ADD #2,R5 ;UPDATE RETURN ADDR
2374 012570 116777 000004 004642 MOVB FNCT,DRPCS ;OUTPUT THE FUNCTION
2375 012576 000205 RTS R5
2376 012600 000000 FNCT: 0
2377
2378 ;RF11 DISK INTERRUPT HANDLER
2379
2380
2381 012602 005067 004726 DKINT: CLR INTERR ;CLEAR THE ERROR FLAG
2382 012606 005777 004626 TST DRPCS ;TEST FOR ERROR
2383 012612 100402 BMI 1$
2384 012614 000167 000322 JMP INTEXT ;JUMP IF NO ERRORS
2385 012620 010667 004710 1$: MOV SP,INTERR ;SET INTERRUPT ERROR FLAG
2386 012624 005767 004700 TST RDERR ;IS THIS THE FIRST ERROR ATTEMPT?
2387 012630 001404 BEQ 2$ ;BRANCH IF YES
2388 012632 032737 000020 177570 BIT #B4,DRSWR ;TYPE ALL ERROR ATTEMPTS?
2389 012640 001524 BEQ DK11 ;BRANCH IF NO
2390 012642 104400 2$: HLT ;STATUS ERROR AFTER INTERRUPT
2391 012644 032777 000002 004566 BIT #B1,DRPCS ;CHECK FOR READ
2392 012652 001012 BNE DELMES ;BRANCH IF WRITTING
2393 012654 004567 166404 JSR R5,PRINTS ;PRINT MESSAGE
2394 012660 016410 MES13 ;GIVE # OF READ ATTEMPT
2395 012662 016767 004642 166672 MOV RDERR,TTY
2396 012670 004767 166462 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2397 012674 010667 004634 3$: MOV SP,INTERR
2398
2399 012700 005067 000274 DELMES: CLR INT1
2400 012704 117767 004540 000266 MOVB DRPCA,INT1 ;GET CYLINDER ADDR
2401 012712 017767 004534 000256 MOV DRPDA,INTO ;GET HEAD AND SECTOR ADDR
2402 012720 042767 160360 000250 BIC #160360,INTO ;CLEAR UNWANTED BITS
2403 012726 032777 000001 004522 BIT #B0,DRPER ;WAS IT AN ADDR ERROR?
2404 012734 001026 BNE REDAC ;BRANCH IF YES
2405 012736 032767 000017 000232 BIT #17,INTO ;IS SECTOR = TO 0
2406 012744 001403 BEQ DECTK ;YES - BRANCH
2407 012746 005367 000224 DEC INTO ;BACK UP COUNT
2408 012752 000417 BR REDAC
2409 012754 132767 000037 000215 DECTK: BITB #37,INTO+1 ;IS HEAD = TO 0
2410 012762 001406 BEQ DECCY ;YES - BRANCH
2411 012764 105367 000207 DECB INTO+1 ;BACK UP HEAD
2412 012770 052767 000011 000200 BIS #11,INTO ;SET UP SECTOR
2413 012776 000405 BR REDAC
2414 013000 012767 011411 000170 DECCY: MOV #11411,INTO
2415 013006 005367 000166 REDAC: DEC INT1
2416 013012

```

```

2417 013012 004567 166246 JSR R5,PRINTS ;PRINT MESSAGE
2418 013016 016424 MES14 ;REPORT CYLINDER ADDR
2419 013020 016767 000154 166534 MOV INT1,TTY
2420 013026 004767 166324 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2421 013032 005067 000142 CLR INT1
2422 013036 116767 000135 000134 MOVB INTO+1,INT1
2423 013044 004567 166214 JSR R5,PRINTS ;PRINT MESSAGE
2424 013050 016441 MES15 ;REPORT HEAD ADDR OF FAILURE
2425 013052 016767 000122 166502 MOV INT1,TTY
2426 013060 004767 166272 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2427 013064 116767 000106 000106 MOVB INTO,INT1
2428 013072 004567 166166 JSR R5,PRINTS ;PRINT MESSAGE
2429 013076 016452 MES16 ;REPORT SEC ADDR OF FAILURE
2430 013100 016767 000074 166454 MOV INT1,TTY
2431 013106 004767 166244 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
2432 013112 032777 001000 004340 DK11: BIT #B9,DRPDS ;IS DRIVE UNSAFE?
2433 013120 001401 BEQ .+4
2434 013122 000000 HALT ;DRIVE UNSAFE
2435 013124 032777 002000 004326 BIT #B10,DRPDS ;SEEK INCOMPLETE?
2436 013132 001403 BEQ INTEXT ;BRANCH IF COMPLETE
2437 013134 112777 000015 004276 MOVB #15,DRPCS ;RECALIBRATE
2438 013142 105777 004272 INTEXT: TSTB DRPCS
2439 013146 100375 BPL .-4 ;WAIT FOR DONE
2440 013150 005777 004304 TST DRPDS
2441 013154 100375 BPL .-4 ;WAIT FOR READY
2442 013156 005767 002642 TST BCKFLG ;DID WE COME FROM BACKGROUND TEST?
2443 013162 001402 BEQ IS ;BRANCH IF NO
2444 013164 012716 016012 MOV #NPRRET,(SP) ;MODIFY RETURN ADDR
2445 013170 010667 171114 IS: MOV SP,INTFLG ;SET INTERRUPT OCCURRED FLAG
2446 013174 000002 RTI
2447
2448 013176 000000 INTO: 0
2449 013200 000000 INT1: 0
2450
2451
2452
2453 ;ROUTINE TO SET UP CYLINDER AND DISK ADDRESS FROM
2454 ;OPERATOR INPUTS DURING CONVERSATION MODE.
2455
2456 013202 032767 000040 004254 OPDSEL: BIT #B5,FLAG ;USE OPERATOR ADDR?
2457 013210 001001 BNE .+4
2458 013212 000207 RTS PC ;NO
2459 013214 016767 004246 004254 MOV SCYL,CYLINDER ;GET CYLINDER ADDR
2460 013222 016767 004244 004250 MOV SSEC,DMA ;GET SECTOR ADDR
2461 013230 116767 004234 004243 MOVB SHED,DMA+1 ;GET HEAD ADDR
2462 013236 000207 RTS PC
2463
2464
2465 ;ROUTINE TO SETUP DISK BUFFERS
2466 ;ADD WORD COUNT TO STARTING DISK ADDRESSES
2467 ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
2468
2469
2470 013240 032767 000040 004216 DISBUF: BIT #B5,FLAG ;DID OPERATOR SUPPLY ADDR?
2471 013246 001401 BEQ .+4
2472 013250 000461 BR BUFEXIT ;OPERATOR DEFINED DISK ADDR

```

```

2473 013252 004767 000520 JSR PC,BLSZ ;DEFINE BLOCK SIZE
2474 013256 016767 004240 004260 MOV BLOCK,WORK1
2475 013264 016767 004210 004250 INCSEC: MOV DMA,WORK ;GET DISK ADDR
2476 013272 042767 177760 004242 BIC #177760,WORK ;MASK OUT SECTOR COUNT
2477 013300 022767 000011 004234 CMP #11,WORK ;CHECK FOR LAST SECTOR
2478 013306 001406 BEQ INCSUR ;CHECK SURFACE
2479 013310 005267 004164 INC DMA ;+1 SECTOR COUNT
2480 013314 005367 004202 DECBLK: DEC BLOCK ;-1 FROM BLOCK COUNT
2481 013320 001432 BEQ CMDAE ;CMP DMA TO RPDA
2482 013322 000760 BR INCSUR ;RECYCLE
2483 013324 042767 000017 004146 INCSEC: BIC #17,DMA ;FETCH ADDRESS
2484 013332 016767 004142 004202 MOV DMA,WORK
2485 013340 042767 160377 004174 BIC #160377,WORK
2486 013346 122767 000023 004167 CMPB #23,WORK+1
2487 013354 001403 BEQ SWSUR ;+1 SURFACE
2488 013356 105267 004117 INCB DMA+1 ;INC HEAD NUMBER
2489 013362 000754 BR DECBLK ;RECYCLE
2490 013364 005067 004110 SWSUR: CLR DMA ;CLEAR THE DISK ADDRESS
2491 013370 005267 004102 INC CYLINDER
2492 013374 022767 000313 004074 CMP #313,CYLINDER ;HAS LAST CYL BEEN EXCEEDED?
2493 013402 001404 BEQ BUFEXIT ;BRANCH IF YES
2494 013404 000743 BR DECBLK
2495 ;COME HERE AFTER DETERMINING THE STARTING ADDR OF THE NEXT
2496 ;TRANSFER. NOW CHECK TO SEE THERE IS ENOUGH ROOM ON THE DISK
2497 ;TO MAKE THE TRANSFER. IF NOT MODIFY THE WORD COUNT FOR THE FINAL
2498 ;OUTPUT.
2499
2500
2501 013406 105767 004052 CMDAE: TSTB FLAG ;CHECK FOR LAST DISK BUFFER
2502 013412 100015 BPL BUFEXIT
2503 013414 005067 004060 BUFEXIT: CLR DMA ;CLEAR ADDRESS BITS
2504 013420 005067 004052 CLR CYLINDER ;CLR CYLINDER REGISTER
2505 013424 062716 000002 ADD #2,(6) ;INC STOCK POINTER
2506 013430 042767 000200 004026 BIC #200,FLAG
2507 013436 016767 004046 004030 MOV SWRDCT,WRDCT
2508 013444 000500 BR EXTDR ;EXIT
2509 013446 005067 004074 BUFEXIT: CLR WORK2 ;CLEAR BLOCK COUNTER
2510 013452 016767 004022 004062 MOV DMA,WORK
2511 013460 016767 004012 004062 MOV CYLINDER,WORK3
2512 013466 042767 160360 004046 BIC #160360,WORK
2513 013474 005267 004046 XINCSEC: INC WORK2 ;INCREMENT BLOCK COUNT
2514 013500 005367 004040 DEC WORK1 ;DECREMENT TOTAL BLOCKS REQUIRED
2515 013504 001460 BEQ EXTDR ;EXIT IF BLOCK COUNT SATISFIED
2516 013506 122767 000011 004026 CMPB #11,WORK ;CHECK THE DISK ADDRESS TO
2517 013514 001403 BEQ XINCSUR ;SEE IF THERE IS ENOUGH ROOM
2518 013516 005267 004020 INC WORK ;TO HANDLE THE OUTPUT REQUESTED
2519 013522 000764 BR XINCSEC
2520 013524 105067 004012 XINCSUR: CLRB WORK
2521 013530 122767 000023 004005 CMPB #23,WORK+1
2522 013536 001403 BEQ IS
2523 013540 105267 003777 INCB WORK+1
2524 013544 000753 BR XINCSEC
2525 013546 005067 003770 IS: CLR WORK
2526 013552 022767 000312 003770 CMP #312,WORK3 ;ARE WE ON THE LAST CYLINDER?
2527 013560 001403 BEQ 2$ ;BRANCH IF YES
2528 013562 005267 003762 INC WORK3
  
```

```

2529 013566 000742          BR      XINCSEC
2530 013570 016767 003752 003676 2$:  MOV    WORK2,WRDCT      ;COME HERE IF THERE IS NOT
2531 013576 000241          CLC                    ;ENOUGH ROOM TO HANDLE THE
2532 013600 006167 003670          ROL    WRDCT           ;REQUESTED OUTPUT. MODIFY THE
2533 013604 006167 003664          ROL    WRDCT           ;WORDCOUNT TO FILL THE REMAINING
2534 013610 006167 003660          ROL    WRDCT           ;SURFACE.
2535 013614 006167 003654          ROL    WRDCT
2536 013620 006167 003650          ROL    WRDCT
2537 013624 006167 003644          ROL    WRDCT
2538 013630 006167 003640          ROL    WRDCT
2539 013634 006167 003634          ROL    WRDCT
2540 013640 052767 000200 003616  EXTDR: BIS    #200,FLAG
2541 013646 000207          RTS    PC              ;EXIT
2542
2543
2544          ;ROUTINE TO SELECT THE DISK UNIT
2545
2546 013650 016767 003610 003664  DSKNOS: MOV    FLAG,WORK      ;FETCH THE FLAG WORD
2547 013656 006067 003660          ROR    WORK
2548 013662 006067 003654          ROR    WORK
2549 013666 000241          CLC
2550 013670 000367 003646          SWAB   WORK
2551 013674 042767 174377 003640  BIC    #174377,WORK      ;MASK THE DISK NUMBER
2552 013702 016777 003634 003530  MOV    WORK,DRPCS      ;LOAD THE ADDRESS IN THE ADDRESS REG
2553 013710 005777 003544          TST    DRPDS           ;IS THE UNIT READY?
2554 013714 100401          BMI    IS              ;BRANCH IF READY
2555 013716 104400          HLT
2556 013720 000205          IS:   RTS    RS        ;SELECTED UNIT NOT READY
2557                                     ;EXIT
2558
2559          ;INITIALIZE THE VECTORS
2560
2561 013722 012767 001112 164104  INIT:  MOV    #ERROR,34      ;SETUP TRAP VECTOR
2562 013730 012767 000340 164100  MOV    #PRI7,36
2563 013736 012767 001004 164064  MOV    #SCOPE$,30      ;SETUP EMT VECTOR
2564 013744 012767 000340 164060  MOV    #PRI7,32
2565 013752 012737 012602 000254  MOV    #DKINT,#VECTOR ;SETUP DISK INTERRUPT VECTOR
2566 013760 012737 000340 000256  MOV    #PRI7,#STATUS
2567 013766 012737 000340 177776  MOV    #PRI7,#PSW      ;LOCKOUT INTERRUPTS
2568 013774 000207          RTS    PC
2569
2570          ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
2571
2572
2573 013776 012767 000377 003516  BLSZ:  MOV    #377,BLOCK      ;DRIVE BLOCK SIZE
2574 014004 016767 003464 003530  MOV    WRDCT,WORK      ;FETCH WORD COUNT
2575 014012 036767 003504 003522  BIT    BLOCK,WORK
2576 014020 001410          BEQ    RORBLK
2577 014022 046767 003474 003512  BIC    BLOCK,WORK      ;SET UP BLOCK OVERFLOW
2578 014030 005267 003466          INC    BLOCK
2579 014034 066767 003462 003500  ADD    BLOCK,WORK
2580 014042 000367 003474          RORBLK: SWAB   WORK
2581 014046 016767 003470 003446  MOV    WORK,BLOCK      ;BLOCK COUNT
2582 014054 000207          RTS    PC              ;EXIT
2583
2584

```

```

2585 ; DETERMINE THE APPROPRIATE ATTENTION BIT FROM
2586 ; THE UNIT NUMBER.
2587
2588 014056 016701 003402 GATTN: MOV FLAG,R1
2589 014062 006001 ROR R1
2590 014064 006001 ROR R1 ; GET UNIT NUMBER
2591 014066 005067 000014 CLR ATTN
2592 014072 042701 177770 BIC #177770,R1 ; ISOLATE UNIT
2593 014076 116167 014110 000002 MOVB ATTNB(R1),ATTN ; GET ATTENTION BIT
2594 014104 000207 RTS PC
2595
2596
2597 014106 000000 ATTN: 0
2598 014110 001 002 004 ATTNB: .BYTE 1,2,4,10,20,40,100,200
2599 014113 010 020 040
2600 014116 100 200
2601 .EVEN
2602
2603
2604
2605
2606 ; ROUTINE TO SELECT DATA PATTERNS FOR TEST
2607
2608 ; ENTER FROM JSR PC PASEL
2609 014120 016700 003360 PASEL: MOV PATNU,R0 ; SET UP PATTERN NUMBER
2610 014124 016767 003344 003410 MOV WRDCT,WORK ; SET UP WORK
2611 014132 012701 017556 MOV #OUTBUF,R1 ; LOC. OF OUTBUFFER
2612 014136 022700 000034 CMP #34,R0 ; TEST FOR RANDOM DATA NUMBER
2613 014142 001423 BEQ RANDOM ; GO GENERATE RANDOM DATA
2614 014144 022700 000032 CMP #32,R0 ; IS THIS PATTERN 15
2615 014150 001406 BEQ PATT32
2616 014152 016021 014556 FILDAT: MOV PATO(0),(1)+ ; FILL BUFFER
2617 014156 005367 003360 DEC WORK ; DEC. WORK COUNT
2618 014162 001373 BNE FILDAT ; LOAD NEXT WORD
2619 014164 000207 RTS PC ; BUFFER FULL
2620 014166 012721 177777 PATT32: MOV #177777,(1)+ ; INSERT ALL ONES PATTERN
2621 014172 005367 003344 DEC WORK
2622 014176 001404 BEQ 1$
2623 014200 005021 CLR (1)+ ; LOAD ZERO PATTERN
2624 014202 005367 003334 DEC WORK ; DECREMENT WORD COUNT
2625 014206 001367 BNE PATT32 ; LOOP IF NOT ZERO
2626 014210 000207 1$: RTS PC ; EXIT
2627
2628 014212 016767 000134 000136 ; RANDOM DATA GENERATOR SUBROUTINE
2629 014220 016767 000130 000132 RANDOM: MOV LONUN,LOSAY
2630 014226 016700 000120 MOV HINUN,HISAV
2631 014232 016704 000116 1$: MOV LONUN,R0 ; SET UP R0 WITH 5 DIGITS LOW
2632 014236 012703 000007 MOV HINUN,R4 ; SET UP R1 WITH 5 DIGITS HIGH
2633 014242 005002 MOV #7,R3 ; SET UP SHIFT COUNT
2634 014244 006300 CLR R2 ; CLEAR R2
2635 014246 006104 2$: ASL R0 ; SHIFT R0 LEFT AND
2636 014250 006102 ROL R4 ; ROTATE CARRY INTO LSB OF R1 INTO
2637 014252 005303 ROL R2 ; ROTATE CARRY OUT OF R1 INTO R2
2638 014254 001373 DEC R3 ; DECREMENT R3
2639 014256 066702 000070 BNE 2$ ; CONTINUE SHIFT LOOP
2640 014262 005504 ADD LONUN,R2 ; ADDN IN NUMBER TO MAKE X 129
ADC R4 ; PROPOGATE CARRY
  
```

2641	014264	066704	000064		ADD	HINUN,R4		;ADDN IN NUMBER TO MAKE X 129
2642	014270	005502			ADC	R2		;PROPOGATE CARRY
2643	014272	062700	001057		ADD	#1057,R0		;ADDN LOW CONSTANT
2644	014276	005504			ADC	R4		;PROPOGATE CARRIES
2645	014300	005502			ADC	R2		;PROPOGATE AGAIN
2646	014302	062704	047401		ADD	#47401,R4		;ADDN HIGH CONSTANT
2647	014306	005502			ADC	R2		;PROPOGATE CARRY
2648	014310	062702	000006		ADD	#6,R2		;ADDN HIGHEST CONSTANT
2649	014314	060200			ADD	R2,R0		;REPRIME R0 WITH HIGH DIGIT
2650	014316	005504			ADC	R4		;PROPOGATE CARRY
2651	014320	010067	000026		MOV	R0,LONUN		;PUT R0 BACK IN LONUN
2652	014324	010021			MOV	R0,(1)+		;HOLD LONUN FOR PROGRAM
2653	014326	005367	003210		DEC	WORK		
2654	014332	001406			BEQ	EXGEN		
2655	014334	010467	000014		MOV	R4,HINUN		;PUT R1 BACK IN HINUN
2656	014340	010421			MOV	R4,(1)+		;HOLD HINUN FOR PROGRAM
2657	014342	005367	003174		DEC	WORK		
2658	014346	001327			BNE	1\$		
2659	014350	000207		EXGEN:	RTS	PC		;RETURN TO PROGRAM
2660	014352	000000		LONUN:	0			
2661	014354	000000		HINUN:	0			
2662	014356	000000		LOSAY:	0			
2663	014360	000000		HISAY:	0			
2664								
2665								
2666	014362	032767	000002	164652	MSG:	BIT	#B1,HLTCT\$;TYPE ENTIRE MESSAGE
2667	014370	001033			BNE	1\$;BRANCH IF NO
2668	014372	004567	164666		JSR	R5,PRINT\$;PRINT MESSAGE
2669	014376	016273			MES8			
2670	014400	004567	164660		JSR	R5,PRINT\$;PRINT MESSAGE
2671	014404	016067			MES2A			
2672	014406	017767	003046	165146	MOV	QRPDS,TTY		
2673	014414	004767	164724		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING QEROS
2674	014420	004567	164640		JSR	R5,PRINT\$;PRINT MESSAGE
2675	014424	016045			MES1A			
2676	014426	017767	003024	165126	MOV	QRPER,TTY		
2677	014434	004767	164704		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING QEROS
2678	014440	004567	164620		JSR	R5,PRINT\$;PRINT MESSAGE
2679	014444	016056			MES2			
2680	014446	017767	002766	165106	MOV	QRPCS,TTY		
2681	014454	004767	164664		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING QEROS
2682	014460	032767	000001	164554	1\$:	BIT	#B0,HLTCT\$;TYPE EXP-REC
2683	014466	001001			BNE	2\$;BRANCH IF YES
2684	014470	000207			RTS	PC		
2685	014472	032767	000002	164542	2\$:	BIT	#B1,HLTCT\$	
2686	014500	001403			BEQ	3\$		
2687	014502	004567	164556		JSR	R5,PRINT\$;PRINT MESSAGE
2688	014506	016465			MES17			
2689	014510				3\$:			
2690	014510	004567	164550		JSR	R5,PRINT\$;PRINT MESSAGE
2691	014514	016505			MES18			
2692	014516	016767	000030	165036	MOV	EXPS,TTY		
2693	014524	004767	164614		JSR	PC,PRINTR		;TYPE LOCATION WITH LEADING QEROS
2694	014530	004567	164530		JSR	R5,PRINT\$;PRINT MESSAGE
2695	014534	016521			MES19			
2696	014536	016767	000012	165016	MOV	RECS,TTY		

2697	014544	004767	164574		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING QEROS
2698	014550	000207			RTS	PC	
2699	014552	000000		EXPS:	0		
2700	014554	000000		RECS:	0		

2701
2702
2703
2704
2705 ;RF11 DATA PATTERNS
2706

2707	014556	163126		PAT0:	163126	
2708	014560	052525		PAT1:	052525	
2709	014562	125252		PAT2:	125252	
2710	014564	031463		PAT3:	031463	
2711	014566	007417		PAT4:	007417	
2712	014570	010421		PAT5:	010421	
2713	014572	021042		PAT6:	021042	
2714	014574	042104		PAT7:	042104	
2715	014576	104210		PAT10:	104210	
2716	014600	167356		PAT11:	167356	
2717	014602	156735		PAT12:	156735	
2718	014604	135673		PAT13:	135673	
2719	014606	073567		PAT14:	073567	
2720	014610	000001		PAT15:	000001	
2721				;PAT16	RANDOM DATA	

2722
2723 ;THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
2724 ;ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 5 OF THE SWITCH
2725 ;REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN
2726 ;ERROR HAS BEEN FOUND AND WILL REPORT UP TO 3 VERIFY ERRORS
2727 ;WITHIN THE SAME INPUT OPERATION.

2728	014612	012767	177775	002720	COMPAR:	MOV	#-3,ERCOUNT	;ERROR RETRY COUNTER
2729	014620	016767	002650	002720		MOV	WRDCT,WORK2	;GET THE WORD COUNT
2730	014626	012767	017556	002656		MOV	#OUTBUF,SAVE	;SET UP OUTBUFFER POINTER
2731	014634	005067	002642			CLR	SWITCH	;CLEAR RANDOM PATTERN FLAG
2732	014640	016767	177512	165034		MOV	LOSAY,LONUM	;GET RANDOM BASE NOS.
2733	014646	016767	177506	165024		MOV	HISAV,HINUM	
2734	014654	022767	000034	002622		CMP	#34,PATNU	;IS THIS RANDOM PATTERN?
2735	014662	001422				BEQ	CMPLP	;BRANCH IF YES
2736	014664	022767	000032	002612		CMP	#32,PATNU	;IS THIS SPECIAL PATTERN?
2737	014672	001037				BNE	CMPLP1	;BRANCH IF NO
2738	014674	005767	002602		CMPLP2:	TST	SWITCH	
2739	014700	001006				BNE	1\$	
2740	014702	012767	177777	177642		MOV	#177777,EXPS	;EXPECT ALL ONES
2741	014710	010667	002566			MOV	SP,SWITCH	;SET THE FLAG
2742	014714	000433				BR	WRDCMP	;GO COMPARE DATA
2743	014716	005067	002560		1\$:	CLR	SWITCH	
2744	014722	005067	177624			CLR	EXPS	;EXPECT ALL ZEROS
2745	014726	000426				BR	WRDCMP	;GO COMPARE DATA
2746	014730	005767	002546		CMPLP:	TST	SWITCH	
2747	014734	001010				BNE	2\$	
2748	014736	004767	164622			JSR	PC,RANDS	;GENERATE TWO RANDOM NOS.
2749	014742	016767	164734	177602		MOV	LONUM,EXPS	;GET EVEN RANDOM WORD
2750	014750	010667	002526			MOV	SP,SWITCH	;SET RANDOM PATTERN FLAG
2751	014754	000413				BR	WRDCMP	
2752	014756	005067	002520		2\$:	CLR	SWITCH	

2753	014762	016767	164712	177562		MOV	HINUM,EXPS	
2754	014770	000405				BR	WRDCMP	
2755	014772	016700	002506		CMPLP1:	MOV	PATNU,RO	
2756	014776	016067	014556	177546		MOV	PAT0(RO),EXPS	
2757	015004	027767	002502	177540	WRDCMP:	CMP	SAVE,EXPS	;COMPARE DATA
2758	015012	001021				BNE	WDERR	;WORD IN ERROR
2759	015014	005367	002526		WRDINC:	DEC	WORK2	;DECREMENT THE WORD COUNT
2760	015020	001415				BEQ	ADAM	;EXIT ROUTINE IF ZERO
2761	015022	062767	000002	002462	BLAD1:	ADD	#2,SAVE	;UPDATE PATTERN ADDRESS
2762	015030	022767	000032	002446		CMP	#32,PATNU	
2763	015036	101362				BHI	WRDCMP	;BRANCH IF STANDARD PATTERN
2764	015040	022767	000034	002436		CMP	#34,PATNU	;IS THIS RANDOM PATTERN
2765	015046	001730				BEQ	CMPLP	;BRANCH IF YES
2766	015050	000711				BR	CMPLP2	;BRANCH IF YES
2767	015052	000754				BR	WRDCMP	;COMPARE NEXT WORD
2768	015054	000207			ADAM:	RTS	PC	;EXIT THIS ROUTINE
2769	015056	005767	167226		WDERR:	TST	INTFLG	;DID INTERRUPT OCCUR YET?
2770	015062	001750				BEQ	WRDCMP	;BRANCH IF NO
2771	015064	017767	002422	177462		MOV	SAVE,RECS	;GET GOOD DATA
2772	015072	010667	002436			MOV	SP,INTERR	;SET ERRORR FLAG
2773	015076	005767	002426			TST	RDERR	;IS THIS THE FIRST READ ERROR?
2774	015102	001404				BEQ	3\$;BRANCH IF YES
2775	015104	032737	000020	177570		BIT	#B4,#SWR	;PRINT ALL RETRY ERRORS?
2776	015112	001550				BEQ	1\$;BRANCH IF NO
2777	015114	104403			3\$:	HLT	+3	;DATA COMPARE ERROR
2778	015116	005067	002400			CLR	BLOCK	;CLEAR THE BLOCK COUNTER
2779	015122	016767	002346	002412		MOV	WRDCT,WORK	;GET THE WORD COUNT
2780	015130	166767	002412	002404		SUB	WORK2,WORK	;DETERMINE DISTANCE OF FAILURE INTO BUFFER
2781	015136	162767	000400	002376	2\$:	SUB	#400,WORK	
2782	015144	100403				BMI	8\$	
2783	015146	005267	002350			INC	BLOCK	;UPDATE BLOCK COUNT FOR EACH 400 WORDS
2784	015152	000771				BR	2\$	
2785	015154	062767	000400	002360	8\$:	ADD	#400,WORK	;RESTORE POSITIVE NUMBER
2786	015162	016767	002312	002354		MOV	DMA,WORK1	;GET HEAD AND SECTOR ADDRESS
2787	015170	016767	002302	002352		MOV	CYLINDER,WORK3	;GET CYLINDER ADDRESS
2788	015176	005767	002320		5\$:	TST	BLOCK	;IS THE BLOCK COUNT ZERO?
2789	015202	001427				BEQ	7\$;BRANCH IF YES
2790	015204	005367	002312			DEC	BLOCK	;DECREMENT BLOCK COUNT
2791	015210	122767	000011	002326		CMPB	#11,WORK1	;DETERMINE THE CYLINDER, HEAD,
2792	015216	001403				BEQ	4\$;AND SECTOR ADDRESSES OF THE
2793	015220	005267	002320			INC	WORK1	;COMPARE ERROR
2794	015224	000764				BR	5\$	
2795	015226	105067	002312		4\$:	CLRB	WORK1	
2796	015232	122767	000023	002305		CMPB	#23,WORK1+1	
2797	015240	001403				BEQ	6\$	
2798	015242	105267	002277			INCB	WORK1+1	
2799	015246	000753				BR	5\$	
2800	015250	005067	002270		6\$:	CLR	WORK1	
2801	015254	005267	002270			INC	WORK3	
2802	015260	000746				BR	5\$	
2803	015262				7\$:			
2804	015262	004567	163776			JSR	R5,PRINT\$;PRINT MESSAGE
2805	015266	016424				MES14		;GIVE CYL ADDR
2806	015270	016767	002254	164264		MOV	WORK3,TTY	
2807	015276	004767	164054			JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2808	015302	005067	002224			CLR	ACNVX	

2809	015306	116767	002233	002216	MOV B	WORK1+1,ACNVX	
2810	015314	004567	163744		JSR	R5,PRINTS	;PRINT MESSAGE
2811	015320	016441			MES15		;GIVE HEAD ADDR
2812	015322	016767	002204	164232	MOV	ACNVX,TTY	
2813	015330	004767	164022		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2814	015334	116767	002204	002170	MOV B	WORK1,ACNVX	
2815	015342	004567	163716		JSR	R5,PRINTS	;PRINT MESSAGE
2816	015346	016452			MES16		;GIVE SECTOR ADDR
2817	015350	016767	002156	164204	MOV	ACNVX,TTY	
2818	015356	004767	163774		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2819	015362	004567	163676		JSR	R5,PRINTS	;PRINT MESSAGE
2820	015366	016312			MES9		
2821	015370	016767	002146	002134	MOV	WORK,ACNVX	;GET WORD COUNT INTO SECTOR
2822	015376	005267	002130		INC	ACNVX	
2823	015402	016767	002124	164152	MOV	ACNVX,TTY	
2824	015410	004767	163742		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2825	015414	004567	163644		JSR	R5,PRINTS	;PRINT MESSAGE
2826	015420	016410			MES13		
2827	015422	016767	002102	164132	MOV	RDERR,TTY	
2828	015430	004767	163722		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2829	015434	032737	000040	177570	1\$: BIT	#B5,#SWR	;CONTINUE COMPARING?
2830	015442	001604			BEQ	ADAM	;BRANCH IF NO
2831	015444	005267	002070		INC	ERCOUNT	;UPDATE ERROR COUNTER
2832	015450	001601			BEQ	ADAM	
2833	015452	000167	177336		JMP	WRDINC	

2834
 2835
 2836
 2837 ;EXTENDED MEMORY EXERCISER
 2838 ;THE PROGRAM DETERMINES HOW MUCH MEMORY
 2839 ;IS ON THE SYSTEM THEN IT
 2840 ;GENERATES A RANDOM BUFFER THAT SIZE
 2841 ;AND WRITES AND WRITE CHECKS THE DATA

2842							
2843	015456	052777	000001	001754	EXTMEN: BIS	#B0,#RPCS	;CLEAR THE DISK
2844	015464	105777	001750		TSTB	#RPCS	
2845	015470	100375			BPL	.-4	
2846	015472	012737	000340	177776	MOV	#PRI7,#PSW	;LOCK UP PRIORITY LEVELS
2847	015500	012767	015550	162276	MOV	#MAXREF,4	;SET UP I/O BUS TRAP
2848	015506	012767	000340	162272	MOV	#PRI7,6	
2849	015514	012767	017446	001770	MOV	#17446,SAVE	;SET UP FOR 4K
2850	015522	005777	001764		EXREF: TST	#SAVE	;REFERENCE MEMORY
2851	015526	022767	157446	001756	CMP	#157446,SAVE	;TEST FOR 28K
2852	015534	001001			BNE	1\$;BRANCH IF LESS THAN 28K
2853	015536	000407			BR	MAXRF1	;LAST REFERENCE MADE TO I/O REG.
2854	015540	062767	020000	001744	1\$: ADD	#20000,SAVE	;SET UP FOR NEXT MEMORY REF.
2855	015546	000765			BR	EXREF	;GO REFERENCE MEMORY

2856
 2857 ;ENTER HERE WHEN I/O BUS ERROR OCCURS
 2858
 2859 015550 162767 020000 001734 MAXREF: SUB #20000,SAVE
 2860 015556 012767 000006 162220 MAXRF1: MOV #6,4 ;RESTORE I/O BUS TRAP
 2861 015564 005067 162216 CLR 6
 2862 015570 005737 000042 TST #42 ;UNDER MONITOR CONTROL?
 2863 015574 001403 BEQ 1\$;BRANCH IF NO
 2864 015576 162767 005670 001706 SUB #3000.,SAVE ;ALLOW ROOM FOR THE MONITOR

```

2865 015604 016767 001702 000070 1$: MOV SAVE, MEMSIZ ;SAVE THE MAXIMUM MEMORY ADDRESS
2866 015612 162767 017556 001672 SUB #OUTBUF, SAVE ;DETERMINE THE BUFFER SIZE
2867 015620 000241 CLC
2868 015622 006067 001664 ROR SAVE ;FORM WORD COUNT
2869 015626 016767 001660 001640 MOV SAVE, WRDCT ;SAVE IT
2870 015634 042767 000001 001650 BIC #80, SAVE ;MAKE ADDRESS EVEN
2871 015642 012767 017556 001656 MOV #OUTBUF, INBUF ;START OF INPUT BUFFER
2872 015650 066767 001636 001650 ADD SAVE, INBUF
2873 015656 000241 CLC
2874 015660 042767 000377 001606 BIC #377, WRDCT ;DETERMINE MAXIMUM WORD COUNT
2875 015666 016767 001602 001614 MOV WRDCT, SWRDCT
2876 015674 012706 000476 MOV #STKPTR-2, SP
2877 015700 000205 RTS RS
2878
2879 015702 000000 MEMSIZ: 0
2880
2881 ;BACKGROUND TEST FOR INTERRUPTS
2882
2883 015704 010667 000114 NPR: MOV SP, BCKFLG ;SET BACKGROUND FLAG
2884 015710 012767 030000 000102 MOV #30000, NPRCNT ;SETUP TIMEOUT COUNTER
2885 015716 012701 016023 MOV #NPR1+1, R1
2886 015722 112711 000200 MOV #200, (R1)
2887 015726 2$:
2888 015726 105421 NEGB (R1)+
2889 015730 105441 NEGB -(R1)
2890 015732 105421 NEGB (R1)+
2891 015734 105441 NEGB -(R1)
2892 015736 105421 NEGB (R1)+
2893 015740 105441 NEGB -(R1)
2894 015742 105421 NEGB (R1)+
2895 015744 105441 NEGB -(R1)
2896 015746 105421 NEGB (R1)+
2897 015750 105441 NEGB -(R1)
2898 015752 105421 NEGB (R1)+
2899 015754 105441 NEGB -(R1)
2900 015756 105421 NEGB (R1)+
2901 015760 105441 NEGB -(R1)
2902 015762 105421 NEGB (R1)+
2903 015764 105441 NEGB -(R1)
2904 015766 102401 BVS 1$
2905 015770 000000 HALT ;ARITHMETIC OPERATION FAILED RUN DIAG
2906 015772 005367 000022 1$: DEC NPRCNT 2$
2907 015776 001353 BNE
2908 016000 104400 HLT ;OPERATION TIMED OUT WAITING FOR INTERRUPT
2909 016002 004567 163256 JSR RS, PRINTS ;PRINT MESSAGE
2910 016006 016562 TIMO
2911 016010 000000 HALT
2912
2913 016012 005067 000006 NPRRET: CLR BCKFLG
2914 016016 000207 RTS PC
2915 016020 000000 NPRCNT: 0
2916 016022 000000 NPR1: 0
2917 016024 000300 BCKFLG: 0
2918
2919
2920

```

2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076

;ERROR MESSAGE HEADERS

016026	005015	047105	020104	MES1: .EVEN
016034	043117	050040	051501	.ASCIZ <15><12>/END OF PASS /
016042	020123	000		
016045	015	051012	042520	MES1A: .ASCIZ <15><12>/RPER= /
016052	036522	000040		
016056	005015	050122	051503	MES2: .ASCIZ <15><12>/RPCS= /
016064	020075	000		
016067	015	051012	042120	MES2A: .ASCIZ <15><12>/RPDS= /
016074	036523	000040		
016100	005015	047527	042122	MES3: .ASCIZ <15><12>/WORD COUNT ISSUED = /
016106	041440	052517	052116	
016114	044440	051523	042525	
016122	020104	020075	000	
016127	015	046412	046505	MES4: .ASCIZ <15><12>/MEMORY ADDRESS= /
016134	051117	020131	042101	
016142	051104	051505	036523	
016150	000040			
016152	005015	040527	052111	MES5: .ASCIZ <15><12>/WAIT 5 SECONDS AND TURN OFF PDP-11 POWER/
016160	032440	051440	041505	
016166	047117	051504	040440	
016174	042116	052040	051125	
016202	020116	043117	020106	
016210	042120	026520	030461	
016216	050040	053517	051105	
016224	000			
016225	015	052012	051505	MES6: .ASCIZ <15><12>/TEST NO /
016232	020124	047516	000040	
016240	005015	047524	040524	MES7: .ASCIZ <15><12>/TOTAL REREADS ON ERROR= /
016246	020114	042522	042522	
016254	042101	020123	047117	
016262	042440	051122	051117	
016270	020075	000		
016273	015	051412	040524	MES8: .ASCIZ <15><12>/STATUS ERROR/
016300	052524	020123	051105	
016306	047522	000122		
016312	005015	047527	042122	MES9: .ASCIZ <15><12>/WORD COUNT INTO SECTOR= /
016320	041440	052517	052116	
016326	044440	052116	020117	
016334	042523	052103	051117	
016342	020075	000		
016345	015	050012	052101	MES10: .ASCIZ <15><12>/PATTERN IN USE= /
016352	042524	047122	044440	

2977	016360	020116	051525	036505	
2978	016366	000040			
2979					
2980	016370	005015	047125	052111	MES11: .ASCIZ <15><12>/UNIT NO. /
2981	016376	047040	027117	000040	
2982					
2983	016404	054503	000114		MES12: .ASCIZ /CYL/
2984					
2985	016410	005015	042522	042101	MES13: .ASCIZ <15><12>/READ NO. /
2986	016416	047040	027117	000040	
2987					
2988	016424	005015	054503	044514	MES14: .ASCIZ <15><12>/CYLINDER= /
2989	016432	042116	051105	020075	
2990	016440	000			
2991					
2992	016441	015	044012	040505	MES15: .ASCIZ <15><12>/HEAD= /
2993	016446	036504	000040		
2994					
2995	016452	005015	042523	052103	MES16: .ASCIZ <15><12>/SECTOR= /
2996	016460	051117	020075	000	
2997					
2998	016465	015	041412	046517	MES17: .ASCIZ <15><12>/COMPARE ERROR/
2999	016472	040520	042522	042440	
3000	016500	051122	051117	000	
3001					
3002	016505	015	042412	050130	MES18: .ASCIZ <15><12>/EXPECTED /
3003	016512	041505	042524	020104	
3004	016520	000			
3005					
3006	016521	015	051012	041505	MES19: .ASCIZ <15><12>/RECEIVED /
3007	016526	044505	042526	020104	
3008	016534	000			
3009	016535	015	047012	020117	MES20: .ASCIZ <15><12>/NO UNITS AVAILABLE/
3010	016542	047125	052111	020123	
3011	016550	053101	044501	040514	
3012	016556	046102	000105		
3013					
3014	016562	005015	051120	041517	TIMO: .ASCIZ <15><12>/PROCESSOR BACKGROUND TEST TIMED OUT/
3015	016570	051505	047523	020122	
3016	016576	040502	045503	051107	
3017	016604	052517	042116	052040	
3018	016612	051505	020124	044524	
3019	016620	042515	020104	052517	
3020	016626	000124			
3021					
3022					:CONVERSATION TEXT
3023					:
3024					:
3025	016630	005015	052123	047101	SPECMES: .ASCIZ <15><12>/STANDARD WORDS TRANSFERRED= /
3026	016636	040504	042122	053440	
3027	016644	051117	051504	052040	
3028	016652	040522	051516	042506	
3029	016660	051122	042105	020075	
3030	016666	000			
3031					
3032	016667	015	042012	052101	CON1: .ASCIZ <15><12>/DATA TEST ONLY? (Y OR N)/

3033	016674	020101	042524	052123	
3034	016702	047440	046116	037531	
3035	016710	024040	020131	051060	
3036	016716	047040	000051		
3037					
3038	016722	005015	052515	052114	CON2: .ASCIZ <15><12>/MULTI DRIVE MODE?(Y OR N)/
3039	016730	020111	051104	053111	
3040	016736	020105	047515	042504	
3041	016744	024077	020131	051117	
3042	016752	047040	000051		
3043					
3044	016756	005015	052516	041115	CON3: .ASCIZ <15><12>/NUMBER OF DRIVES 1 TO 10 OCTAL?/
3045	016764	051105	047440	020106	
3046	016772	051104	053111	051505	
3047	017000	030440	052040	020117	
3048	017006	030061	047440	052103	
3049	017014	046101	000077		
3050					
3051	017020	005015	044127	041511	CON4: .ASCIZ <15><12>/WHICH DRIVE?/
3052	017026	020110	051104	053111	
3053	017034	037505	000		
3054					
3055	017037	015	047412	052120	CON5: .ASCIZ <15><12>/OPTIONAL WORD COUNT? (Y OR N)/
3056	017044	047511	040516	020114	
3057	017052	047527	042122	041440	
3058	017060	052517	052116	020077	
3059	017066	054450	047440	020122	
3060	017074	024516	000		
3061					
3062	017077	015	046012	047105	CON6: .ASCIZ <15><12>/LENGTH? (1 TO SWRDCT)/
3063	017104	052107	037510	024040	
3064	017112	020061	047524	051440	
3065	017120	051127	041504	024524	
3066	017126	000			
3067					
3068	017127	015	051412	040524	CON7: .ASCIZ <15><12>/STARTING HEAD?/
3069	017134	052122	047111	020107	
3070	017142	042510	042101	000077	
3071					
3072	017150	005015	047504	054440	CON7A: .ASCIZ <15><12>/DO YOU WISH TO SELECT THE DISK TEST ADDR?(Y OR N)/
3073	017156	052517	053440	051511	
3074	017164	020110	047524	051440	
3075	017172	046105	041505	020124	
3076	017200	044124	020105	044504	
3077	017206	045523	052040	051505	
3078	017214	020124	042101	051104	
3079	017222	024077	020131	051117	
3080	017230	047040	000051		
3081					
3082	017234	005015	052123	051101	CON7B: .ASCIZ <15><12>/STARTING SECTOR?/
3083	017242	044524	043516	051440	
3084	017250	041505	047524	037522	
3085	017256	000			
3086					
3087	017257	015	051412	040524	CON7C: .ASCIZ <15><12>/STARTING CYLINDER?/
3088	017264	052122	047111	020107	

```
3089 017272 054503 044514 042116
3090 017300 051105 000077
3091
3092 017304 005015 050117 044524 CON8: .ASCIZ <15><12>/OPTIONAL DATA PATTERN NO. ?/
3093 017312 047117 046101 042040
3094 017320 052101 020101 040520
3095 017326 052124 051105 020116
3096 017334 047516 037456 000
3097
3098 017341 015 053412 044522 CON9: .ASCIZ <15><12>/WRITE?(Y OR N)/
3099 017346 042524 024077 020131
3100 017354 051117 047040 000051
3101
3102 017362 005015 051127 052111 CON10: .ASCIZ <15><12>/WRITE CHECK?(Y OR N)/
3103 017370 020105 044103 041505
3104 017376 037513 054450 047440
3105 017404 020122 024516 000
3106
3107 017411 015 051012 040505 CON11: .ASCIZ <15><12>/READ?(Y OR N)/
3108 017416 037504 054450 047440
3109 017424 020122 024516 000
3110
3111 017431 015 042412 042116 END: .ASCIZ <15><12>/END/
3112 017436 000
3113 017440 .EVEN
```

3114
 3115
 3116
 3117
 3118
 3119
 3120 017440 176714
 3121 017442 176715
 3122 017444 176716
 3123 017446 176720
 3124 017450 176722
 3125 017452 176724
 3126 017454 176725
 3127 017456 176712
 3128 017460 176710
 3129 017462 176723
 3130
 3131
 3132
 3133
 3134 017464 000000
 3135 017466 000000
 3136 017470 000000
 3137 017472 000000
 3138 017474 000000
 3139 017476 000000
 3140 017500 000000
 3141 017502 000000
 3142 017504 000000
 3143 017506 000000
 3144 017510 000000
 3145 017512 000000
 3146 017514 000000
 3147 017516 000000
 3148 017520 000000
 3149 017522 000000
 3150 017524 000000
 3151 017526 000000
 3152 017530 000000
 3153 017532 000000
 3154 017534 000000
 3155 017536 000000
 3156 017540 000000
 3157 017542 000000
 3158 017544 000000
 3159 017546 000000
 3160 017550 000000
 3161 017552 000000
 3162 017554 000000
 3163
 3164
 3165 017556 000000
 3166
 3167 000001

.....
 ;DISK I/O REGISTERS
 RPCS: 176714
 RPCS1: 176715
 RPWC: 176716
 RPBA: 176720
 RPCA: 176722
 RPDA: 176724
 RPDA1: 176725
 RPER: 176712
 RPDS: 176710
 RPCA1: 176723

;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

;DEDICATED REGISTERS

FLAG: 0
 SCYL: 0
 SHED: 0
 SSEC: 0
 WRDCT: 0
 CYLINDER: 0
 DMA: 0
 SWITCH: 0
 PATNU: 0
 BUF: 0
 SWRDCT: 0
 SAVE: 0
 SAVI: 0
 PASS: 0
 DSKNOR: 0
 BLOCK: 0
 PASSC: 0
 INBUF: 0
 RDERR: 0
 ACNVX: 0
 INTERR: 0
 TESTNO: 0
 ERCOUNT: 0
 WORK: 0
 WORK1: 0
 WORK2: 0
 WORK3: 0
 WORK4: 0
 CYLA: 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

CUTBUF:0
 .END

CON7	017127	1126	3068#											
CON7A	017150	1111	3072#											
CON7B	017234	1134	3082#											
CON7C	017257	1118	3087#											
CON8	017304	1143	3092#											
CON9	017341	1156	3098#											
CYLA	017554	1273*	1274*	1275*	1276*	1358*	1359*	1360*	1361*	3162#				
CYLIND	017476	1038*	1248*	1253	1290	1292	1312*	1315*	1374*	1377	1423	1468	1481	1521*
		1524	1533	1535*	1546*	1559	1561	1577*	1582*	1606*	1754*	1771*	1794*	1848*
		2029*	2160*	2239*	2286*	2368	2459*	2491*	2492	2504*	2511	2787	3139#	
DATAT	010332	1219	1226	2003	2020#									
DATP	010434	2035#	2124											
DATTES	002620	1061	1079#	1085										
DECBK	013314	2480#	2489	2494										
DECCY	013000	2410	2414#											
DECTK	012754	2406	2409#											
DELMES	012700	2392	2399#											
DISBUF	013240	1443	1500	2049	2063	2120	2470#							
DISPLA=	177570	683#												
DKINT	012602	2033	2381#	2565										
DKI1	013112	2389	2432#											
DMA	017500	1039*	1373*	1379	1382	1384*	1386*	1410	1472	1477	1479*	1483	1490*	1520*
		1607*	1753*	1772*	1773*	1793*	1847*	2030*	2168*	2238*	2287*	2367	2460*	2461*
		2475	2479*	2483*	2484	2488*	2490*	2503*	2510	2786	3140#			
DREAD	010606	2052	2065#											
DSKDR	002526	1063#	1069	1072										
DSKNOR	017520	1037*	1073*	1074*	1076*	1077*	1181*	1183	1186*	1188*	1190*	1191*	1193	2244
		3148#												
DSKNOS	013650	1247	1345	1519	1605	2279	2366	2546#						
DSKRD	010626	2069#	2104	2110										
EMTVEC=	000030	673#												
END	017431	2346	3111#											
EOPTST	006700	1753#	1811											
ERCOUN	017540	2728*	2831*	3156#										
ERRFLG	001240	788	790*	839*	842#									
ERROR	001112	819#	2561											
ERRORS	001246	840*	845#											
ERRPC	001252	829	847#											
ERRVEC=	000004	668#												
ESH	010616	2067#	2118	2121										
EXGEN	014350	2654	2659#											
EXMFLG	003546	1201	1210#											
EXPS	014552	1292*	1406*	1410*	1411	1413*	1416*	1417	1423*	1426*	1427	1429*	1430	1481*
		1483*	1561*	1727*	1739*	1806*	1880*	1956*	1988*	2326*	2337*	2692	2699#	2740*
		2744*	2749*	2753*	2756*	2757								
EXREF	015522	2850#	2855											
EXTDR	013646	2508	2515	2541#										
EXTEND	010260	1999#	2010											
EXTMEN	015456	1034	2843#											
EXTPP	012070	2249	2256	2262#										
EXTRP	010306	1916	1998	2006#										
EXTTRP	010312	1911	2007#											
EXTTST	007620	1911#												
EXTT1	010100	1961#												
FILDAT	014152	2616#	2618											
FLAG	017464	1036*	1043*	1056*	1062*	1089*	1115*	1150*	1154*	1160*	1167*	1174*	1175	1192*

		1456*	1488*	1514*	1519*	1525*	1551*	1600*	1605*	1617*	1625*	1632*	1650*	1657*
		1697*	1706*	1715*	1762*	1780*	1795*	1828*	1849*	1862*	1890*	1928*	1942*	1964*
		1978*	2021*	2040*	2054*	2070*	2113*	2141*	2170*	2179*	2190*	2217*	2251*	2279*
		2294*	2303*	2305*	2340*	2345*	2366*	2372*	2373*	2375*	2393*	2417*	2423*	2428*
		2556*	2668*	2670*	2674*	2678*	2687*	2690*	2694*	2804*	2810*	2815*	2819*	2825*
		2877*	2909*											
SANHT	005206	1472*	1476											
SAVE	017512	2730*	2757	2761*	2771	2849*	2850	2851	2854*	2859*	2864*	2865	2866*	2868*
		2869	2870*	2872	3145*									
SAVES	001036	796*	912	974										
SAVI	017514	3146*												
SCOPE =	104000	724*	1308	1370	1442	1499	1545	1573	1667	1744	1812	1884	1906	1960
		1997	2048	2062	2119	2178	2187	2223	2302					
SCOPE\$	001004	786*	2563											
SCYL	017466	1123*	2459	3135*										
SEABUF	004502	1375*	1444											
SHED	017470	1131*	2461	3136*										
SP =	%000006	650*	792*	796	797*	798*	799*	800*	801*	802*	807	808	809	810
		811	812	813	858*	859*	863	880*	907	1010*	1032*	1414	2007*	2276*
		2358*	2385	2397	2444*	2445	2741	2750	2772	2876*	2883			
SPECME	016630	1048	3025*											
SRO =	177572	708*	1913	1923*	2006*									
SSEC	017472	1139*	2460	3137*										
START	002336	774	1031*											
STATUS=	000256	1028*	1246*	2034*	2566*									
STKPTR=	000500	686*	1032	2007	2276	2358	2876							
SWITCH	017502	2731*	2738	2741*	2743*	2746	2750*	2752*	3141*					
SWR =	177570	682*	786	819	822	826	836	852	870	873	1041	1210	1212	1319
		1502	1585	1813	2002	2028*	2042	2056	2072	2074	2133	2172	2181	2192
		2197	2228	2388	2775	2829								
SWRUCT	017510	1049	1103	1107*	1108	2032	2507	2875*	3144*					
SWSUR	013364	2487	2490*											
T =	000020	658*												
TBITVE=	000014	670*												
TESTNO	017536	1240*	1243	1340*	1343	1513*	1516	1599*	1602	1827*	1830	2020*	2023	2140*
		2143	3155*											
TIMO	016562	2910	3014*											
TKB =	177562	679*	955											
TKS =	177560	678*												
TKSR	002772	1095	1109*											
TPB =	177566	681*	865*	955*										
TPS =	177564	680*	856	866	956									
TRAPVE=	000034	674*												
TSTTBL	003612	1216	1221*											
TTY	001562	833*	887*	889*	891*	910*	1049*	1208*	1243*	1343*	1516*	1602*	1634*	1659*
		1830*	2023*	2115*	2143*	2219*	2253*	2342*	2395*	2419*	2425*	2430*	2672*	2676*
		2680*	2692*	2696*	2806*	2812*	2817*	2823*	2827*					
		718*	1920	1921	1922									
UP =	000000	655*												
V =	000002	1027*	1245*	2033*	2565*									
VECTOR=	000254	1096*	1102	1106										
WCCON	002710	2758	2769*											
WDERR	015056	1103*	1104*	1105	1401*	1414*	1424	2150*	2155*	2157*	2158	2160	2242*	2243*
WORK	017542	2244	2315*	2317*	2318*	2319*	2320*	2321	2475*	2476*	2477	2484*	2485*	2486
		2510*	2512*	2516	2518*	2520*	2521	2523*	2525*	2546*	2547*	2548*	2550*	2551*
		2552	2574*	2575	2577*	2579*	2580*	2581	2610*	2617*	2621*	2624*	2653*	2657*

MOV8	832	865	876	879	903	947	1018	1183	1253	1294	1302	1409	1410	1422	1436
	1493	1539	1567	1771	1773	2105	2209	2280	2328	2368	2374	2400	2422	2427	2437
NEG	2461	2593	2809	2814	2886										
NEGB	2371														
	2898	2889	2890	2891	2892	2893	2894	2895	2896	2897	2898	2899	2900	2901	2902
	2903														
NOP	2259	2260	2261												
RESET	1031	2257													
ROL	887	889	891	918	919	986	987	988	999	1004	1005	1006	1007	1076	1077
	1087	1088	1153	1190	1191	1215	1671	2532	2533	2534	2535	2536	2537	2538	2539
	2635	2636													
ROLB	888	890	892												
ROR	993	994	1205	1206	1837	2027	2317	2318	2547	2548	2589	2590	2868		
RTI	791	793	841	1325	2446										
RTS	855	864	872	875	908	937	964	1013	1021	2375	2458	2462	2541	2556	2568
	2582	2594	2619	2626	2659	2684	2698	2768	2877	2914					
SUB	831	1070	1835	2093	2780	2781	2859	2864	2866						
SWAB	992	997	1002	2319	2550	2580									
TRAP	722														
TST	788	819	836	1015	1068	1101	1179	1184	1193	1263	1271	1277	1281	1297	1305
	1309	1356	1362	1366	1393	1397	1404	1424	1439	1449	1460	1470	1474	1496	1529
	1542	1555	1570	1574	1583	1614	1621	1629	1637	1641	1647	1662	1668	1701	1710
	1719	1733	1759	1766	1777	1799	1804	1843	1853	1866	1876	1913	1933	1946	1968
	1982	2098	2108	2111	2125	2202	2212	2215	2292	2322	2382	2386	2440	2442	2553
	2738	2746	2769	2773	2788	2850	2862								
TSTB	856	861	866	893	896	945	956	977	1259	1303	1350	1391	1437	1447	1458
	1494	1527	1540	1553	1568	1619	1627	1652	1699	1708	1717	1764	1782	1797	1851
	1864	1892	1931	1944	1966	1980	2096	2106	2210	2290	2296	2307	2324	2438	2501
	2844														
WAIT	2046	2060	2078	2176	2185	2196									
.ABS	637														
.ASCIZ	846	847	967	968	2925	2928	2931	2934	2936	2941	2946	2955	2958	2965	2969
	2975	2980	2983	2985	2988	2992	2995	2998	3002	3006	3009	3014	3025	3032	3038
	3044	3051	3055	3062	3068	3072	3082	3087	3092	3098	3102	3107	3111		
.BLKW	909	966													
.BYTE	2598														
.ENABL	1025														
.END	3167														
.ENDC	836														
.EVEN	849	2601	2704	2924	3113										
.IF	835														
.LIST	635	768	772	1025											
.MACR	720	751	752	753	754	755	756	757	758	759	760	762	763	764	765
	766														
.MACRO	761														
.MLIST	636	768	1025												
.PAGE	1816	2265	2361												
.REM	19														
.REPT	768	2887													
.SBTTL	1232	1328	1505	1589	1816	2011	2136	2265	2361						
.TITLE	638	1025													
.WORD	1388	1457	1526	1552	1618	1626	1651	1698	1707	1716	1763	1781	1796	1850	1863
	1891	1929	1943	1965	1979	2041	2055	2071	2171	2180	2191	2295	2306		

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*.DZRPFB.SEQ/SOL/CRF/PAGNUM+DZRPFB
RUN-TIME: 10 20 4 SECONDS
RUN-TIME RATIO: 192/36=5.2
CORE USED: 10K (19 PAGES)

