

RP11-C/RP02

MULTIDRIVE
MD-11-DZRPB-B

EP DZRPB B DL A

OCT 1976

COPYRIGHT 1976

digital

FICHE 1 OF 1

Made in U.S.A.

1	Introduction
2	Hardware
3	Software
4	Installation
5	Operation
6	Maintenance
7	Appendix A
8	Appendix B
9	Appendix C
10	Appendix D
11	Appendix E
12	Appendix F
13	Appendix G
14	Appendix H
15	Appendix I
16	Appendix J
17	Appendix K
18	Appendix L
19	Appendix M
20	Appendix N
21	Appendix O
22	Appendix P
23	Appendix Q
24	Appendix R
25	Appendix S
26	Appendix T
27	Appendix U
28	Appendix V
29	Appendix W
30	Appendix X
31	Appendix Y
32	Appendix Z

RP11
MULTI DRIVE DIAGNOSTIC
PROGRAM

.REM *

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRPB-B-D
PRODUCT NAME: RP11 MULTI DRIVE DIAGNOSTIC
DATE CREATED: JUNE 15, 1973
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: JOE STUBBLEBINE

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

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

1. ABSTRACT

THIS PROGRAM WILL TEST UP TO EIGHT RPO2 DRIVES ON AN RP11 DISK CONTROLLER. BASICALLY, THE PROGRAM WILL SEEK TO A RANDOM ADDRESS AND THEN WRITE AND READ RANDOM DATA. WHILE DATA IS BEING TRANSFERRED, SEEK OPERATIONS WILL BE IN PROGRESS ON THE OTHER DRIVES. THE PURPOSE OF THE TEST IS TO CHECK FOR ANY INTERACTION ON THE BUS WHILE TRYING TO KEEP ALL THE DRIVES BUSY.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 STANDARD FAMILY PROCESSOR
RP11 DISK CONTROLLER WITH UP TO EIGHT RPO2 DISK DRIVES

2.2 STORAGE

4K OF STORAGE IS REQUIRED TO RUN THIS TEST

2.3 PRELIMINARY PROGRAMS

DZRPE DISKLESS DIAGNOSTIC
DZRPF DISK RELIABILITY DIAGNOSTIC

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1

E01

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 4
DIRPGB.P11

144
145

4.2 STARTING ADDRESS

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
 201

THE PROGRAM SHOULD 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 (SEE SEC. 5.1.1)
4. PRESS START.
5. THE PROGRAM WILL LOOP UNTIL STOPPED
6. DUE TO THE RANDOM NATURE OF THE PROGRAM THERE IS NO MEANINGFULL PASSCOUNT. IT IS RECOMMENDED THAT THE PROGRAM RUN AT LEAST HALF AN HOUR.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT STARTING ADDRESS 200, THE SETTING OF THE SWITCHES WILL DETERMINE WHICH UNITS ARE TO BE TESTED.

5.1.1 SWITCH SETTING ARE:

- SW<15>=1.....HALF ON ERROR
- SW<14>NOT USED
- SW<13>=1.....INHIBIT PRINTOUT
- SW<12>NOT USED
- SW<11>NOT USED
- SW<10>=1.....BELL ON ERROR
- SW<07> THRU SW<00>=1.....SELECT UNIT FOR TEST

SW<00> CORRESPONDS TO UNIT 0
 SW<07> CORRESPONDS TO UNIT 7

5.2 SUBROUTINE ABSTRACTS

5.2.1 HLT

GO1

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 6
DZRPGB.P11

202
203

THIS ROUTINE IS ENTERED UPON DETECTION OF AN ERROR.

204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258

IT WILL TYPE THE PC OF THE ERROR AND ADDITIONAL ERROR INFORMATION. THIS ROUTINE TESTS FOR, HALT ON ERROR, INHIBIT TYPEOUTS, AND RINGS THE BELL.

5.2.2 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ANY UNEXPECTED TRAPS. THESE 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. ALSO, THE CONTENTS OF THE SELECTED CYLINDER, HEAD AND SECTOR ADDRESS ARE TYPED. BY REFERRING TO THE LISTING, ADDITIONAL INFORMATION CAN BE FOUND REGARDING THE CAUSE OF THE ERROR IN THE COMMENT FIELD. WHEN APPROPRIATE, ADDITIONAL INFORMATION IS TYPED OUT, SUCH AS THE EXPECTED AND RECEIVED RESULTS OF AN OPERATION. ALL INFORMATION IS IN OCTAL.

ERROR MESSAGE FORMAT

- 1. PC= ADDRESS OF FAILURE
- UNIT UNIT WHICH FAILED
- RPDS= CONTENTS OF RPDS
- RPER= CONTENTS OF RPER
- RPCS= CONTENTS OF RPCS
- CYLINDER SELECTED CYLINDER
- HEAD SELECTED HEAD
- SECTOR SELECTED SECTOR
- EXPECTED EXPECTED DATA
- RECEIVED RECEIVED DATA

7.0 RESTRICTIONS

SINCE THIS IS AN INTERACTION TEST, THERE IS NO LOOPING ON ERRORS.

8.0 MISCELLANEOUS

259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314

8.1 EXECUTION TIME

DUE TO THE RANDOM NATURE OF THE PROGRAM THERE IS NO MEANINGFUL PASS COUNT. IT IS RECOMMENDED THAT THE PROGRAM SHOULD RUN FOR HALF AN HOUR.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500.

8.3 ERROR INFORMATION

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

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

9.0 PROGRAM DESCRIPTION

WHEN STARTED THE PROGRAM WILL RESTORE THE HEADS FOR EACH OF THE SELECTED UNITS. THEN THE FOLLOWING SEQUENCE IS GONE THRU FOR EACH OF THE SELECTED UNITS. FIRST, A RANDOM DISK SURFACE ADDRESS IS GENERATED AND A SEEK IS ISSUED. THEN A RANDOM BUFFER IS SELECTED AND FILLED WITH RANDOM DATA. A SECTOR IS THEN WRITTEN, READ BACK AND COMPARED. THIS SEQUENCE IS THEN LOOPED UPON. DUE TO THE DIFFERENCE IN SEEK TIMES, WHICH DEPENDS ON THE RANDOM DISK ADDRESS SELECTED, ALL UNITS ARE EXERCISED IN A RANDOM SELECTION. WHILE DATA IS BEING TRANSFERRED, SEEK OPERATIONS ARE IN PROGRESS.

%

```

000000' 012706 000000G          START:  MOV      #STKPTR,SP      ;SET STACK POINTER
000004' 012737 000340 000000G      MOV      #340,2#PSW      ;LOCK UP INTERRUPTS
000012' 012767 000000G 000034      MOV      #ERROR,34      ;SETUP ERROR TRAP
000020' 012767 000000G 000036      MOV      #PRI7,36
000026' 012767 000000G 000030      MOV      #SCOPE$,30     ;SETUP SCOPE TRAP

```

J01

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 9
DZRPGB.P11

```

315 000034' 012767 000000G 000032      MOV      #PRI7,32
316 000042' 012737 002336' 000254      MOV      #DSKINT,2#VECTOR      ;SET UP DISK INTERRUPT VECTOR
317 000050' 012737 000340 000256      MOV      #340,2#STATUS
318 000056' 005000      CLR      RO
319 000060' 005060 002632'      CLRTAB: CLR      DEVTBL(RO)      ;CLEAR THE DEVICE TABLE
320 000064' 005720      TST      (RO)+
321 000066' 022700 000200      CMP      #128,RO
322 000072' 001372      BNE      CLRTAB
323 000074' 005737 000042      TST      2#42      ;UNDER MONITOR CONTROL?
324 000100' 001424      BEQ      5$      ;BRANCH IF NO
325 000102' 005000      CLR      RO      ;CLEAR MODIFIER
326 000104' 005001      CLR      R1
327 000106' 012777 000001 002742 7$:      MOV      #1,2RPCS      ;CLEAR THE CONTROLLER
328 000114' 110177 002740      MOV      R1,2RPCS1      ;SELECT UNIT
329 000120' 005777 002750      TST      2RPCS      ;IS UNIT READY?
330 000124' 100403      BMI      6$      ;BRANCH IF YES
331 000126' 052760 000000G 002632'      BIS      #B15,DEVTBL(RO)      ;SET UNIT UNAVAILABLE BIT
332 000134' 062700 000020 5$:      ADD      #16.,RO      ;UPDATE MODIFIER
333 000140' 005201      INC      R1      ;UPDATE UNIT NUMBER
334 000142' 032701 000000G      BIT      #B3,R1      ;ALL UNITS TESTED?
335 000146' 001757      BEQ      7$      ;BRANCH IF NO
336 000150' 000420      BR      8$
337 000152' 012701 000001 5$:      MOV      #1,R1
338 000156' 005000      CLR      RO
339 000160' 030137 000000G 2$:      BIT      R1,2#SWR      ;TEST THE SWITCH REGISTER
340 000164' 001003      BNE      1$      ;TO DETERMINE WHICH UNITS
341 000166' 052760 000000G 002632'      BIS      #B15,DEVTBL(RO)      ;TO TEST. IF THE UNIT IS UNAVAILABLE
342 000174' 062700 000020 1$:      ADD      #16.,RO      ;SET BIT 15 IN THE DEVICE TABLE
343 000200' 000241      CLC
344 000202' 006101      ROL      R1
345 000204' 032701 000000G      BIT      #B8,R1      ;HAVE ALL UNITS BEEN SCANNED?
346 000210' 001763      BEQ      2$      ;NO-BRANCH
347 000212' 000005 8$:      RESET      ;CLEAR THE SYSTEM
348 000214' 004567 001656      JSR      R5,EXTMEN      ;DETERMINE AMOUNT OF CORE
349 000220' 005067 002606      CLR      UNIT      ;INITIALIZE POINTER
350 000224' 005067 002604      CLR      PASSCT
351 000230' 005005      CLR      R5
352 000232' 032765 000000G 002632' 4$:      BIT      #B15,DEVTBL(R5)      ;IS UNIT AVAILABLE?
353 000240' 001002      BNE      3$      ;BRANCH IF NO
354 000242' 004767 000136      JSR      PC,HOME      ;DO A HOME SEEK
355 000246' 005267 002560 3$:      INC      UNIT      ;UPDATE UNIT
356 000252' 062705 000020      ADD      #16.,R5      ;UPDATE TABLE POINTER
357 000256' 032767 000000G 002546      BIT      #B3,UNIT      ;HAVE ALL UNITS BEEN HOMED?
358 000264' 001762      BEQ      4$      ;NO-BRANCH
359 000266' 005067 002540      LOOP: CLR      UNIT
360 000272' 005005      CLR      R5
361 000274' 032765 000000G 002632' MAIN:      BIT      #B15,DEVTBL(R5)      ;IS THE UNIT AVAILABLE?
362 000302' 001004      BNE      1$      ;BRANCH IF NO
363 000304' 016504 002632'      MOV      DEVTBL(R5),R4
364 000310' 004774 000550'      JSR      PC,2JMPTBL(R4)      ;PERFORM FUNCTION IN JMPTBL
365 000314' 005267 002512 1$:      INC      UNIT      ;UPDATE UNIT
366 000320' 062705 000020      ADD      #16.,R5      ;UPDATE TABLE POINTER
367 000324' 032767 000000G 002500      BIT      #B3,UNIT      ;HAVE ALL UNITS BEEN SCANNED?
368 000332' 001760      BEQ      MAIN      ;NO BRANCH
369 000334' 005267 002474      INC      PASSCT      ;INCREMENT ITERATION COUNTER
370 000340' 016737 002470 000000G      MOV      PASSCT,2#SWR      ;DISPLAY COUNT

```

K01

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 10
 DZRPGB.P11

```

371 000346' 005737 000042          TST      Q#42          ;UNDER MONITOR CONTROL?
372 000352' 001413          BEQ      MEXIT1       ;BRANCH IF NO
373 000354' 022767 005000 002452    CMP      #5000,PASSCT ;IS PASS COMPLETE?
374 000362' 001007          BNE      MEXIT1       ;BRANCH IF NO
375 000264' 013701 000042          MOV      Q#42,R1     ;GET RETURN ADDRESS
376 000370' 000005          RESET
377 000372' 004711          MEXIT:  JSR      PC,(R1) ;RETURN TO MONITOR
378 000374' 000240          NOP
379 000376' 000240          NOP
380 000400' 000240          NOP
381 000402' 000731          MEXIT1: BR      LOOP          ;LOOP
382
383          ;THIS ROUTINE WILL SEEK HOME THE PACK WHOSE
384          ;ADDRESS IS IN UNIT.
385
386 000404' 116777 002422 002446    HOME:  MOVB     UNIT,QRPDS1 ;LOAD THE UNIT #
387 000412' 005777 002456          TST      QRPDS       ;IS THE UNIT READY?
388 000416' 100401          BMI     S$           ;BRANCH IF READY
389 000420' 000000G          HLT
390 000422' 112777 000015 002426    S$:    MOVB     #15,QRPCS ;UNIT IS NOT READY
391 000430' 012704 000025          MOV      #25,R4     ;DO A HOME SEEK
392 000434' 005304          1$:    DEC      R4
393 000436' 001376          BNE     1$           ;WAIT FOR SEEK TO START
394 000440' 032777 000000G 002426    BIT     #810,QRPDS  ;IS SEEK UNDER WAY SET
395 000446' 001001          BNE     2$           ;YES
396 000450' 000000G          HLT     ;SEEK UNDERWAY DID NOT SET
397 000452' 016704 002354          2$:    MOV      UNIT,R4
398 000456' 005067 000054          CLR     ATTNB
399 000462' 116467 000540' 000046    MOVB    ATTN(R4),ATTNB ;DETERMINE ATTENTION RESPONSE
400 000470' 005000          CLR     RO
401 000472' 005200          7$:    INC     RO
402 000474' 001376          BNE     7$
403 000476' 005200          6$:    INC     RO
404 000500' 036777 000032 002366    BIT     ATTNB,QRPDS ;TIME OUT ATTENTION BIT
405 000506' 001003          BNE     3$           ;DID ATTENTION SET?
406 000510' 005700          TST     RO           ;BRANCH IF YES
407 000512' 001371          BNE     6$           ;DID IT TIME OUT?
408 000514' 000000G          HLT     ;BRANCH IF NO
409          ;ATTENTION BIT DID NOT SET
410 000516' 005777 002334          3$:    TST     QRPDS     ;ANY DEVICE STATUS ERRORS?
411 000522' 100001          BPL     4$           ;NO-BRANCH
412 000524' 000000G          HLT     ;DEVICE STATUS ERROR AFTER HOME COMMAND
413 000526' 046777 000004 002340    4$:    BIC     ATTNB,QRPDS ;CLEAR ATTENTION BIT
414 000534' 000207          RTS     PC           ;EXIT
415
416 000536' 000000          ATTNB:  0           ;CONTAINS ATTENTION BIT FOR CURRENT UNIT
417 000540' 001 002 004          ATTN:  .BYTE 1,2,4,10,20,40,100,200
418 000543' 010 020 040
419 000546' 100 200
420
421          .EVEN
422          ;THIS TABLE DETERMINES WHERE CONTROL WILL GO FOR ANY
423          ;PARTICULAR UNIT. THE FIRST WORD OF THE DEVICE TABLE
424          ;IS USED AS A MODIFIER FOR A JSR INDIRECT INTO
425          ;THE FOLLOWING TABLE.
426

```

```

427 000550' 000560'          JMPTBL: SEEN          ;SEEK A RANDOM CYLINDER
428 000552' 001040'          SEKCK          ;SEE IF SEEK IS COMPLETE
429 000554' 001354'          WRITE         ;WRITE RANDOM DATA
430 000556' 001530'          READD         ;READ AND VERIFY RANDOM DATA
431
432          ;THIS ROUTINE WILL GENERATE A RANDOM CYLINDER
433          ;ADDRESS AND ISSUE A SEEK TO IT. THE FUNCTION
434          ;PCOUNTER IN THE DEVICE TABLE WILL BE UPDATED TO
435          ;CHECK FOR THE ATTENTION BIT.
436
437 000560' 016565 002636' 002634' SEEK:  MOV     DEVTBL+4(R5),DEVTBL+2(R5) ;SET UP CYLINDER FROM ADDRESS
438 000566' 116777 002240 002264     MOVVB  UNIT,ARPCSI          ;SELECT THE UNIT
439 000574' 000000G          1$:  RAND          ;GENERATE A RANDOM NUMBER
440 000576' 016767 000000G 002234     MOV     LONUM,WORK1
441 000604' 016767 000000G 002230     MOV     HINUM,WORK2
442 000612' 042767 177400 002220     BIC     #177400,WORK1
443 000620' 022767 000312 002212     CMP     #312,WORK1          ;IS NUMBER TOO LARGE?
444 000626' 002762          BLT     1$          ;BRANCH IF YES
445 000630' 026765 002204 002634'     CMP     WORK1,DEVTBL+2(R5)
446 000636' 001756          BEQ     1$
447 000640' 016765 002174 002636'     MOV     WORK1,DEVTBL+4(R5)          ;SAVE CYLINDER ADDRESS
448 000646' 016765 002170 002640'     MOV     WORK2,DEVTBL+6(R5)          ;USE A RANDOM DATA BASE
449 000654' 000000G          RAND          ;GENERATE A RANDOM NUMBER
450 000656' 016767 000000G 002154     MOV     LONUM,WORK1
451 000664' 016767 000000G 002150     MOV     HINUM,WORK2
452 000672' 042767 177760 002140     BIC     #177760,WORK1
453 000700' 022767 000011 002132     CMP     #11,WORK1          ;GENERATE A RANDOM SECTOR
454 000706' 002003          BGE     2$
455 000710' 162767 000010 002122     SUB     #8,WORK1
456 000716' 042767 177740 002116     BIC     #177740,WORK2          ;GENERATE A RANDOM TRACK
457 000724' 022767 000023 002110     CMP     #23,WORK2
458 000732' 002003          BGE     3$
459 000734' 162767 000014 002100     SUB     #14,WORK2
460 000742' 116767 002074 002071     3$:  MOVVB  WORK2,WORK1+1          ;MERGE TRACK AND SECTOR ADDR
461
462 000750' 016765 002064 002642'     MOV     WORK1,DEVTBL+10(R5)          ;STORE RANDOM DISK ADDRESS
463 000756' 005065 002644'     CLR     DEVTBL+12(R5)          ;CLEAR TIMEOUT COUNTER
464 000762' 116577 002636' 002076     MOVVB  DEVTBL+4(R5),ARPCA          ;LOAD CYLINDER ADDRESS
465 000770' 016577 002642' 002072     MOV     DEVTBL+10(R5),ARPCDA          ;LOAD TRACK AND SECTOR
466 000776' 112777 000011 002052     MOVVB  #11,ARPCS          ;INITIATE A SEEK
467 001004' 012704 000025          MOV     #25,R4
468 001010' 005304          4$:  DEC     R4          ;WAIT FOR SEEK TO START
469 001012' 001376          BNE     4$
470 001014' 032777 000000G 002052     BIT     #810,ARPCDS          ;IS THE SEEK UNDERWAY?
471 001022' 001001          BNE     5$          ;YES-BRANCH
472 001024' 000000G          HLT          ;SEEK UNDERWAY DID NOT SET
473 001026' 005265 002632'     5$:  INC     DEVTBL(R5)          ;MODIFY FUNCTION POINTER TO
474 001032' 005265 002632'     INC     DEVTBL(R5)          ;CHECK FOR SEEK COMPLETE
475 001036' 000207          RTS     PC          ;EXIT
476
477          ;THIS ROUTINE IS ENTERED AFTER A SEEK HAS BEEN ISSUED.
478          ;IT CHECKS THE ATTENTION FLAG - IF CLEAR IT UPDATES THE
479          ;TIMEOUT COUNTER AND CHECKS IT. IF SET IT VERIFIES
480          ;THE SEEK FUNCTIONED PROPERLY.
481
482 001040' 016704 001766     SEKCK: MOV     UNIT,R4

```

```

483 001044' 116467 000540' 177464      MOVB  ATTN(R4),ATTNB ; DETERMINE ATTENTION BIT
484 001052' 036777 177460 002014      BIT   ATTNB,ARPCD   ; TEST FOR ATTENTION FLAG
485 001060' 001014          BNE   1$          ; BRANCH IF SET
486 001062' 005265 002644'          INC   DEVTBL+12(R5) ; UPDATE TIMEOUT COUNTER
487 001066' 022765 005000 002644'      CMP   #5000,DEVTBL+12(R5) ; DID OPERATION TIMEOUT?
488 001074' 101005          BHI   2$          ; NOT YET-BRANCH
489 001076' 116777 001730 001754      MOVB  UNIT,ARPCD   ; SELECT UNIT
490 001104' 000000G        HLT                   ; SEEK TIMED OUT
491 001106' 000440          BR    4$          ;
492 001110' 000207          RTS   PC           ; EXIT
493 001112' 146777 177420 001754      BICB  ATTNB,ARPCD   ; CLEAR ATTENTION FLAG
494 001120' 116777 001706 001732      MOVB  UNIT,ARPCD   ; SELECT UNIT
495 001126' 032777 000000G 001740      BIT   #B10,ARPCD   ; IS SEEK UNDERWAY CLEAR?
496 001134' 001402          BEQ   3$          ; IF YES-BRANCH
497 001136' 000000G        HLT                   ; SEEK UNDERWAY DID NOT CLEAR
498 001140' 000423          BR    4$          ;
499 001142' 005777 001710          3$:  TST   ARPCD       ; ARE THERE ANY DEVICE STATUS ERRORS?
500 001146' 100002          BPL   5$          ; BRANCH-NO ERRORS
501 001150' 000000G        HLT                   ; DEVICE STATUS ERRORS
502 001152' 000416          BR    4$          ;
503 001154' 117767 001716 001446      5$:  MOVB  ARPCD1,RECS ; GET CURRENT CYLINDER ADDRESS
504 001162' 026765 001442 002636'      CMP   RECS,DEVTBL+4(R5) ; DOES IT MATCH CYLINDER REQUESTED?
505 001170' 001427          BEQ   6$          ; YES-BRANCH
506 001172' 000000G        HLT                   ; CURRENT CYLINDER AND CYL REQUESTED DID NOT COMPARE
507 001174' 000000G        PRINT
508 001176' 001262'        MES10
509 001200' 000000G        DUMP  DEVTBL+4(R5)
510 001202' 000000G        PRINT
511 001204' 001311'        MES11
512 001206' 000000G        DUMP
513 001210' 032777' 000000G 001656  4$:  BIT   #B11,ARPCD   ; SEEK INCOMPLETE?
514 001216' 001411          BEQ   10$         ; BRANCH IF NO
515 001220' 112777 000015 001630      MOVB  #15,ARPCD   ; ISSUE HOME COMMAND
516 001226' 105777 001624          TSTB  ARPCD       ; WAIT FOR DONE
517 001232' 100375          BPL   -4          ;
518 001234' 005777 001634          TST   ARPCD       ; WAIT FOR UNIT READY
519 001240' 100375          BPL   -4          ;
520 001242' 005065 002632'          10$: CLR   DEVTBL(R5)  ; CLEAR FUNCTION POINTER
521 001246' 000207          RTS   PC           ; EXIT
522 001250' 005265 002632'          6$:  INC   DEVTBL(R5)
523 001254' 005265 002632'          INC   DEVTBL(R5)  ; UPDATE FUNCTION POINTER
524 001260' 000207          RTS   PC           ; EXIT
525
526 001262' 005015 042522 052521  MES10: .ASCIZ  <15><12>'REQUESTED CYLINDER= '
527 001270' 051505 042524 020104
528 001276' 054503 044514 042116
529 001304' 051105 020075          000
530 001311' 015 041412 051125  MES11: .ASCIZ  <15><12>'CURRENT CYLINDER ADDR REGISTER= '
531 001316' 042522 052116 041440
532 001324' 046131 047111 042504
533 001332' 020122 042101 051104
534 001340' 051040 043505 051511
535 001346' 042524 036522 000040
536
537
538

```

.EVEN

; THIS ROUTINE WILL WRITE A RANDOM SECTOR ON

```

539                                     ;A RANDOM TRACK. THE CYLINDER HAS ALREADY
540                                     ;BEEN SELECYED BY THE SEEK ROUTINE.
541
542 001354' 004767 000634 WRITE: JSR PC,RANADR ;GENERATE A RANDOM BUFFER ADDR
543 001360' 012767 000400 001450 MOV #400,WORK
544 001366' 016701 001456 MOV BUFF,R1
545 001372' 004767 000334 JSR PC,RANDAT ;GENERATE A RANDOM PATTERN
546 001376' 116777 001430 001454 MOV# UNIT,ARPCSI ;SELECT THE UNIT
547 001404' 032777 000000G 001462 BIT #B15,ARPDS ;IS THE SELECTED UNIT READY
548 001412' 001003 BNE 1$ ;YES-BRANCH
549 001414' 000000G HLT ;SELECTED UNIT NOT READY
550 001416' 000167 000100 JMP WRTER ;START SEQUENCE OVER
551
552 001422' 012777 177400 001432 1$: MOV #-400,ARPC ;SETUP WORD COUNT REGISTER
553 001430' 016777 001414 001426 MOV BUFF,ARPBA ;SETUP BUS ADDR REGISTER
554 001436' 016577 002636' 001422 MOV DEVTBL+4(R5),ARPCA ;SET CYLINDER ADDR
555 001444' 016577 002642' 001416 MOV DEVTBL+10(R5),ARPDA ;SETUP DISK ADDR.
556 001452' 005067 001374 CLR INT ;CLEAR INTERRUPT FLAG
557 001456' 012737 000000G 000000G MOV #PRI4,ARPSW ;ALLOW INTERRUPT
558 001464' 005067 001356 CLR INTERR ;CLEAR ERROR FLAG
559 001470' 112777 000113 001360 MOV# #113,ARPCS ;INITIATE WRITE WITH INTERRUPT
560 001476' 004767 000606 JSR PC,WAIT ;TIMEOUT THE OPERATION
561 001502' 005767 001340 TST INTERR ;ANY ERRORS?
562 001506' 001005 BNE WRTER ;BRANCH IF YES
563 001510' 005265 002632' INC DEVTBL(R5) ;UPDATE FUNCTION POINTER TO READ
564 001514' 005265 002632' INC DEVTBL(R5)
565 001520' 000403 BR READD
566 001522' 005065 002632' WRTER: CLR DEVTBL(R5) ;RESTORE FUNCTION POINTER
567 001526' 000207 RTS PC ;EXIT
568
569                                     ;READ AND VERIFY THE DATA WRITTEN
570
571 001530' 116777 001276 001322 READD: MOV# UNIT,ARPCSI ;SELECT THE UNIT
572 001536' 032777 000000G 001330 BIT #B15,ARPDS ;IS THE SELECTED UNIT READY?
573 001544' 001003 BNE 1$ ;YES-BRANCH
574 001546' 000000G HLT ;SELECTED UNIT NOT READY
575 001550' 000167 000150 JMP RDCNT
576 001554' 012777 177400 001300 1$: MOV #-400,ARPC ;LOAD WORD COUNT REGISTER
577 001562' 016777 001262 001274 MOV BUFF,ARPBA ;LOAD BUS ADDR REGISTER
578 001570' 062777 001000 001266 ADD #1000,ARPBA
579 001576' 016577 002636' 001262 MOV DEVTBL+4(R5),ARPCA ;SET CYLINDER ADDR
580 001604' 016577 002642' 001256 MOV DEVTBL+10(R5),ARPDA ;SETUP DISK ADDR.
581 001612' 005067 001234 CLR INT ;CLEAR INTERRUPT FLAG
582 001616' 005067 001224 CLR INTERR ;CLEAR ERROR FLAG
583 001622' 112777 000117 001226 MOV# #117,ARPCS ;INITIATE READ WITH INTERRUPT
584 001630' 004767 000454 JSR PC,WAIT ;TIMEOUT THE OPERATION
585 001634' 032777 000000G 001214 BIT #B14,ARPCS ;ANY HARD ERRORS?
586 001642' 001030 BNE RDCNT ;BRANCH IF YES
587 001644' 016701 001200 MOV BUFF,R1
588 001650' 016702 001174 MOV BUFF,R2
589 001654' 005003 CLR R3
590 001656' 062701 001000 ADD #1000,R1 ;START OF PATTERN BUFFER
591 001662' 022122 3$: CMP (R1)+,(R2)+ ;COMPARE DATA
592 001664' 001006 BNE 2$ ;BRANCH-DATA DID NOT COMPARE
593 001666' 005203 INC R3 ;INCREMENT COUNTER
594 001670' 022703 000400 CMP #400,R3 ;HAS BUFFER BEEN SCANNED

```

```

595 001674 001372 BNE 35 ;BRANCH-NC
596 001676 000167 000022 JMP RDCNT
598 001702 016267 177776 000716 25: MOV -2(R2),EXPS
599 001710 016167 177776 000712 MOV -2(R1),RECS
600 001716 000000G PRINT
601 001720 003205 MES12
602 001722 000001G HLT +1 ;DATA DID NOT VERIFY
603 001724 005065 002632 RDCNT: CLR DEVTBL(R5) ;INITIATE FUNCTION POINTER
604 001730 000207 RTS PC ;EXIT
605
606 ;GENERATE A RANDOM PATTERN
607
608 001732 016567 002640 000132 RANDAT: MOV DEVTBL+6(R5),RAND1 ;GET RANDOM BASE
609 001740 016567 002642 000126 MOV DEVTBL+10(R5),RAND2
610 001746 016700 000120 MOV RAND1,R0
611 001752 016704 000116 MOV RAND2,R4
612 001756 012703 000007 RANDA1: MOV #7,R3 ;SETUP SHIFT COUNT
613 001762 005002 CLR R2
614 001764 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
615 001766 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R0 INTO R4
616 001770 006102 ROL R2 ;ROTATE CARRY OUT OF R4 INTO R2
617 001772 005303 DEC R3 ;DECREMENT R3
618 001774 001373 BNE SHIFT ;CONTINUE LOOP
619 001776 066700 000070 ADD RAND1,R0 ;ADD IN # TO MAKE X129
620 002002 005504 ADC R4 ;PROPOGATE CARRY
621 002004 066704 000064 ADD RAND2,R4 ;ADD IN # TO MAKE X129
622 002010 005502 ADC R2 ;PROPOGATE CARRY
623 002012 062700 001057 ADD #1057,R0 ;ADD LOW CONSTANT
624 002016 005504 ADC R4 ;PROPOGATE CARRY
625 002020 005502 ADC R2 ;PROPOGATE CARRY
626 002022 062704 047401 ADD #47401,R4 ;ADD HIGH CONSTANT
627 002026 005502 ADC R2
628 002030 062702 000006 ADC #6,R2
629 002034 060200 ADD R2,R0 ;REPRIME R0 WITH HIGH DIGIT
630 002036 005504 ADC R4
631 002040 010067 000026 MOV R0,RAND1
632 002044 010021 MOV R0,(R1)+ ;STORE DATA IN BUFFER
633 002046 005367 000764 DEC WORK
634 002052 001406 BEQ EXGEN
635 002054 010467 000014 MOV R4,RAND2
636 002060 010421 MOV R4,(R1)+ ;STORE DATA IN BUFFER
637 002062 005367 000750 DEC WORK
638 002066 001333 BNE RANDA1 ;FILL ENTIRE BUFFER
639 002070 000207 EXGEN: RTS PC ;EXIT
640
641 002072 000000 RAND1: 0
642 002074 000000 RAND2: 0
643
644 ;THIS ROUTINE DETERMINES THE TOTAL AMOUNT OF AVAILABLE
645 ;CARE WITHOUT USING MEMORY MANAGEMENT.
646
647 002076 012737 000000G 000000G EXTMEN: MOV #PRI7,0PSW ;LOCKUP PRIORITY LL.EL
648 002104 012767 002154 000004 MOV #MAXREF,4 ;SETUP IO BUS TRAP
649 002112 012767 000000G 000006 MOV #PRI7,6
650 002120 012767 017446 000064 MOV #17446,SAVE ;START WITH 4k
  
```

```

651 002126' 005777 000060 EXREF: TST      JSAVE          ;REFERENCE MEMORY
652 002132' 022767 157446 000052      CMP      #157446,SAVE ;TEST FOR 28K
653 002140' 001001          BNE      1$          ;BRANCH IF LESS THAN 28K
654 002142' 000407          BR       MAXRF1
655 002144' 062767 020000 000040 1$:      ADD      #20000,SAVE ;SETUP FOR NEXT REFERENCE
656 002152' 000765          BR       EXREF
657
658                                     ;ENTER HERE WHEN IO BUS ERROR OCCURS
659
660 002154' 162767 020000 000030 MAXREF: SUB      #20000,SAVE
661 002162' 012767 000006 000004 MAXRF!: MOV      #6,4          ;RESTORE IO BUS TRAY
662 002170' 005067 000006          CLR      6
663 002174' 005737 000042          TST      #42          ;UNDER MONITOR CONTROL?
664 002200' 001403          BEQ      1$          ;BRANCH IF NO
665 002202' 162767 005670 000002      SUB      #3000.,SAVE ;ADJUST TOP OF CORE
666 002210' 000205          1$:      RTS      R5          ;EXIT-SAVE=MAXIMUM MEMORY
667 002212' 000000          SAVE:   0          ;HIGHEST AVAILABLE LOCATION
668
669                                     ;GENERATE A RANDOM BUFFER ADDRESS
670
671 002214' 016704 177772      RANADR: MOV      SAVE,R4
672 002220' 162704 003312'      SUB      #ENDP,R4          ;DETERMINE BUFFER SIZE
673 002224' 162704 002000          SUB      #2000,R4          ;ALLOW ROOM FOR DATA
674 002230' 000000G          RAND          ;GENERATE A RANDOM #
675 002232' 016767 000000G 000600      MOV      LONUM,WORK1
676 002240' 042767 000000G 000572      BIC      #80,WORK1          ;MAKE NUMBER EVEN
677 002246' 012703 100000          MOV      #100000,R3
678 002252' 020467 000562      2$:      CMP      R4,WORK1          ;ENSURE THAT THE RANDOM
679 002256' 101005          BHI      1$          ;ADDRESS FITS WITHIN AVAILABLE
680 002260' 040367 000554          BIC      R3,WORK1          ;MEMORY
681 002264' 000241          CLC
682 002266' 006003          ROR
683 002270' 000770          BR       R3
684 002272' 062767 003312' 000540 1$:      ADD      #ENDP,WORK1
685 002300' 016767 000534 000542      MOV      WORK1,BUFF          ;SAVE BUFFER START ADDR.
686 002306' 000207          RTS      PC          ;EXIT
687
688                                     ;TIMEOUT THE OCCURANCE OF AN INTERRUPT
689
689 002310' 005000      WAIT:   CLR      R0
690 002312' 005200      2$:      INC      R0
691 002314' 005767 000532          TST      INT          ;HAS INTERRUPT OCCURED?
692 002320' 001005          BNE      1$          ;YES-BRANCH
693 002322' 005700          TST      R0          ;HAS OPERATION TIMED OUT?
694 002324' 001372          BNE      2$          ;NO-BRANCH
695 002326' 000000G          HLT          ;UNIT TIMED OUT ON READ OR WRITE
696 002330' 005267 000512          INC      INTERR          ;SET ERROR FLAG
697 002334' 000207      1$:      RTS      PC          ;EXIT
698
699                                     ;ENTERED UPON A DEVICE INTERRUPT. THIS ROUTINE
700                                     ;WILL CHECK FOR AND REPORT DEVICE ERRORS
701
702 002336' 032777 000000G 000512      DSKINT: BIT      #B15,DZRPCS ;WHERE THER ANY ERRORS?
703 002344' 001402          BEQ      1$          ;BRANCH-NO ERRORS
704 002346' 000167 000064          JMP      DSKER          ;REPORT ERROR
705 002352' 016703 000472      1$:      MOV      BUFF,R3
706 002356' 062703 001000          ADD      #1000,R3

```


707	002362'	022765	000004	002632'		CMP	#4,DEVTBL(R5)	: IS THIS A WRITE?
708	002370'	001402				BEQ	2\$: BRANCH IF YES
709	002372'	062703	001000			ADD	#1000,R3	
710	002376'	020377	000462		3\$:	CMP	R3,ARPBA	: DID THE BUS ADDR TERMINATE PROPERLY?
711	002402'	001413				BEQ	2\$: YES-BRANCH
712	002404'	000000G				HLT		: CONTENTS OF RPBA INCORRECT
713	002406'	000000G				PRINT		
714	002410'	003232'				MES13		
715	002412'	000000G				PRINT		
716	002414'	003262'				MES18		
717	002416'	000000G				DUMP	R3	
718	002420'	000000G				PRINT		
719	002422'	003276'				MES19		
720	002424'	000000G				DUMP	ARPBA	
721	002425'	005267	000414			INC	INTERR	: SET ERROR FLAG
722	002432'	000167	000006		2\$:	JMP	EXTINT	
723	002436'	000000G			DSKER:	HLT		: REPORT INTERRUPT DISK ERROR
724	002440'	005267	000402			INC	INTERR	: SET ERROR FLAG
725	002444'	052767	000000G	000400	EXTINT:	BIS	#B0,INT	
726	002452'	032777	000000G	000414	1\$:	BIT	#B1\$,ARPDS	: IS THE UNIT READY
727	002460'	001774				BEQ	1\$: NO-WAIT
728	002462'	000002				RTI		: EXIT INTERRUPT
729								
730								
731	002464'	032767	000000G	000000G	MSG:	BIT	#B1,HLTCTS	: TYPE ENTIRE MSG?
732	002472'	001041				BNE	1\$: BRANCH IF NO
733	002474'	000000G				PRINT		
734	002476'	003100'				MES1		
735	002500'	000000G				SDUMP	UNIT	
736	002502'	000000G				PRINT		
737	002504'	003132'				MES2A		
738	002506'	017767	000352	000000G		MOV	ARPDS,TTY	
739	002514'	004767	000500G			JSR	PC,PRINTR	
740	002520'	000000G				PRINT		
741	002522'	003110'				MES1A		
742	002524'	000000G				DUMP	ARPER	
743	002526'	000000G				PRINT		
744	002530'	003121'				MES2		
745	002532'	000000G				DUMP	ARPCS	
746	002534'	000000G				PRINT		
747	002536'	003143'				MES3		
748	002540'	000000G				SDUMP	DEVTBL+4(R5)	
749	002542'	000000G				PRINT		
750	002544'	003160'				MES4		
751	002546'	005067	000272			CLR	WORK3	
752	002552'	116567	002643'	000264		MOVB	DEVTBL+11(R5),WORK3	
753	002560'	000000G				SDUMP	WORK3	
754	002562'	000000G				PRINT		
755	002564'	003172'				MES5		
756	002566'	116567	002642'	000250		MOVB	DEVTBL+10(R5),WORK3	
757	002574'	000000G				SDUMP	WORK3	
758	002576'	032767	000000G	000000G	1\$:	BIT	#B0,HLTCTS	: TYPE EXPECTED - RECEIVED?
759	002604'	001001				BNE	2\$: BRANCH IF YES
760	002606'	000207				RTS	PC	
761	002610'	000000G			2\$:	PRINT		
762	002612'	003262'				MES18		

```

Q 763 002614' 000000G          DUMP   EXPS
  764 002616' 000000G          PRINT
  765 002620' 003276'          MES'9
Q 766 002622' 000000G          DUMP   RECS
  767 002624' 000207          RTS    PC
  768 002626' 000000
  769 002630' 000000
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790 002632' 000000          DEVTBL: 0          ;EVEN          ;UNIT 0 SLOT
  791          002652'          ;=DEVTBL+20
  792 002652' 000000          UNIT1: 0          ;UNIT 1 SLOT
  793          002672'          ;=UNIT1+20
  794 002672' 000000          UNIT2: 0          ;UNIT 2 SLOT
  795          002712'          ;=UNIT2+20
  796 002712' 000000          UNIT3: 0          ;UNIT 3 SLOT
  797          002732'          ;=UNIT3+20
  798 002732' 000000          UNIT4: 0          ;UNIT 4 SLOT
  799          002752'          ;=UNIT4+20
  800 002752' 000000          UNIT5: 0          ;UNIT 5 SLOT
  801          002772'          ;=UNIT5+20
  802 002772' 000000          UNIT6: 0          ;UNIT6 SLCT
  803          003012'          ;=UNIT6+20
  804 003012' 000000          UNIT7: 0          ;UNIT 7 SLOT
  805          003032'          ;=UNIT7+20
  806
  807          ;RP11 CONSTANTS-MEMORY ASSIGNMENTS
  808 003032' 000000          UNIT: 0          ;CURRENT UNIT UNDER TEST
  809 003034' 000000          PASSCT: 0          ;COUNTS EACH PASS THRU 8 UNITS
  810 003036' 000000          WORK: 0          ;TEMPORARY STORAGE AREA
  811 003040' 000000          WORK1: 0
  812 003042' 000000          WORK2: 0
  813 003044' 000000          WORK3: 0
  814 003046' 000000          INTERR: 0          ;INTERRUPT ERROR FLAG
  815 003050' 000000          BUFF: 0          ;STARTING ADDRESS OF BUFFER
  816 003052' 000000          INT: 0          ;INTERRUPT FLAG
  817 003054' 000000          FLAG: 0          ;FLAG WORD
  818          ;DISK IO REGISTERS

```

EXP\$:
 RECS:
 ;DEVTBL IS A TABLE CONTAINING SLOTS FOR EACH OF EIGHT
 ;POSSIBLE UNITS. DURING THE OPERATION OF THE PROGRAM
 ;RS IS USED AS A MODIFIER TO POINT INTO THE TABLE TO
 ;SELECT THE PROPER UNIT. EACH UNIT SLOT CONTAINS
 ;EIGHT ENTRIES(WORD)
 ;1 FUNCTION POINTER
 ;0=SEEK
 ;2=SEEK IN PROGRESS
 ;4=WRITE
 ;6=READ
 ;IF NEGATIVE-UNIT IS NOT TESTED
 ;2 CYLINDER FROM ADDRESS-INDICATES PREVIOUS CYLINDER POSITION
 ;3 CYLINDER TO ADDRESS-ADDRESS OF THE SEEK COMMAND
 ;4 RANDOM BASE FOR PATTERN GENERATION
 ;5 RANDOM TRACK AND SECTOR ADDRESS
 ;6 CYLINDER SEEK TIMEOUT COUNTER
 ;7 SPARE
 ;8 SPARE

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 22
 DZRPGB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

UNITS	002752R	800#	801														
UNIT6	002772R	802#	803														
UNIT7	003012R	804#	805														
VECTOR=	000254	316#	828#														
WAIT	002310R	560	584	689#													
WORK	003036R	543#	633#	637#	810#												
WORK1	003040R	440#	442#	443	445	447	450*	452*	453	455*	460*	462	675*	676*			
		678	680#	684#	685	811#											
WORK2	003042R	441*	448	451*	456*	457	459*	460	812#								
WORK3	003044R	751*	752*	756*	813#												
WRITE	001354R	429	542#														
WATER	001522R	550	562	566#													
.	= 003314R	517	519	791#	793#	795#	797#	799#	801#	803#	805#						

ADC	620	622	624	625	627	630										
ADD	332	342	356	366	578	590	619	621	623	626	628	629	655	684	706	
ASL	709															
BEQ	614															
BGE	324	335	346	358	368	372	446	496	505	514	634	664	703	708	711	
BHI	454	458														
BIC	488	679														
BICB	413	442	452	456	676	680										
BIS	493															
BIT	331	341	725													
BLT	334	339	345	352	357	361	367	394	404	470	484	495	513	547	572	
BMI	585	702	726	731	758											
BNE	444															
BPL	330	388														
BR	322	340	353	362	374	393	395	402	405	407	469	471	485	548	562	
CLC	573	586	592	595	618	638	653	692	694	732	759					
CLR	411	500	517	519												
CMP	336	381	491	498	502	565	654	656	683							
DEC	343	681														
INC	318	319	325	326	338	349	350	351	359	360	398	400	463	520	556	
JMP	558	566	581	582	589	603	613	662	689	751						
JSR	321	373	443	445	453	457	487	504	591	594	652	678	707	710		
MOV	392	468	617	633	637											
MOV	333	355	365	369	401	403	473	474	486	522	523	563	564	593	690	
MOV	696	721	724													
MOV	550	575	596	704	722											
MOV	348	354	364	377	542	545	560	584	739							
MOV	310	311	312	313	314	315	316	317	327	337	363	370	375	391	397	
MOV	437	440	441	447	448	450	451	462	465	467	482	543	544	552	553	
MOV	554	555	557	576	577	579	580	587	588	598	599	608	609	610	611	
MOV	612	631	632	635	636	647	648	649	650	661	671	675	677	685	705	
MOV	738															
MOV	328	386	390	399	438	460	464	466	483	489	494	503	515	546	559	
MOV	571	583	752	756												
MOV	378	379	380													
MOV	347	376														
MOV	344	615	616													
MOV	682															
MOV	728															
MOV	414	475	492	521	524	567	604	639	666	686	697	760	767			
MOV	455	459	660	665	672	673										
MOV	320	323	329	371	387	406	410	499	518	561	651	663	691	693		
MOV	516															
MOV	526	530	833	835	837	839	841	844	846	848	852	856	858			
MOV	417															
MOV	861															
MOV	421	536	789													
MOV	19															

. ABS. 000000 000
 003314 001

% ERRORS DETECTED: 12

K02

.MAIN. MACY11 27(732) 16-SEP-76 15:47 PAGE 25
DZRPGB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

% DEFAULT GLOBALS GENERATED: 25

*.DZRPGB.SEG/SOL/CRF/PAGNUM+DZRPGB
RUN-TIME: 27.1 SECONDS
RUN-TIME RATIO: 154/11=13.4
CORE USED: 6K (11 PAGES)

