

# RP11-C/RP03

MULTIDISK

MD-11-DZRPC-B

EP DZRPC B-DL A

OCT 1976

COPYRIGHT ©1976

**digital**

FICHE 1 OF 1

Made in U.S.A.







**E01**

RP11C MULTI DRIVE DIAGNOSTIC  
DZRPCB.M01

MACY11 27(732) 16-SEP-76 15:29 PAGE 4

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

RPI1C MULTI DRIVE DIAGNOSTIC  
DZRPCB.MD1

MACY11 27(732) 16-SEP-76 15:29 PAGE 6

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.

%

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

315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
  
000001  
000002  
000004  
000010  
000020  
000340  
000300  
000240  
000200  
000140  
000100  
000040  
  
000004  
000010  
000014  
000020  
000024  
000030  
000034  
  
177776  
177560  
177562  
177564  
177566  
177570  
177570  
  
000500  
  
100000  
040000  
020000  
010000  
004000  
002000  
001000

;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

;GENERAL REGISTER ASSIGNMENTS

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

;STATUS REGISTER (PSW) BIT ASSIGNMENTS

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

;VECTOR ADDRESSES

ERRVEC=4  
RESVEC=10  
TBITVEC=14  
IOTVEC=20  
PFVEC=24  
EMTVEC=30  
TRAPVEC=34

;ERROR VECTOR  
;RESERVED INST VECTOR  
;T BIT VECTOR  
;IOT TRAP VECTOR  
;POWER FAIL VECTOR  
;EMT VECTOR  
;TRAP VECTOR

;REGISTER ADDRESSES

PSW=177776  
TKS=177560  
TKB=177562  
TPS=177564  
TPB=177566  
SWR=177570  
DISPLAY=177570

;PROCESSOR STATUS REGISTER  
;KEYBOARD CSR  
;ADDR OF KEYBOARD BUFFER  
;TELEPRINTER CSR  
;TELEPRINTER BUFFER  
;CONSOLE SWITCH REGISTER  
;CONSOLE DISPLAY REGISTER

;INITIAL STACK POINTER

STKPTR=500

;PROGRAM STACK POINTER

;BIT ASSIGNMENTS

B15=100000  
B14=40000  
B13=20000  
B12=10000  
B11=4000  
B10=2000  
B9=1000

371 000400  
 372 000200  
 373 000100  
 374 000040  
 375 000020  
 376 000010  
 377 000004  
 378 000002  
 379 000001

B8=400  
 B7=200  
 B6=100  
 B5=40  
 B4=20  
 B3=10  
 B2=4  
 B1=2  
 B0=1

;MEMORY MANAGEMENT REGISTER ASSIGNMENTS

380  
 381  
 382  
 383 177572  
 384 172340  
 385 172342  
 386 172344  
 387 172346  
 388 172350  
 389 172352  
 390 172354  
 391 172356  
 392 172300  
 393 172302  
 394 172304  
 395 172306  
 396 172310  
 397 172312  
 398 172314  
 399 172316  
 400 000006  
 401 000000  
 402  
 403

SRO=177572  
 KIPAR0=172340  
 KIPAR1=172342  
 KIPAR2=172344  
 KIPAR3=172346  
 KIPAR4=172350  
 KIPAR5=172352  
 KIPAR6=172354  
 KIPAR7=172356  
 KIPDR0=172300  
 KIPDR1=172302  
 KIPDR2=172304  
 KIPDR3=172306  
 KIPDR4=172310  
 KIPDR5=172312  
 KIPDR6=172314  
 KIPDR7=172316  
 RW=6  
 UP=00

;INSTRUCTION EQUATES

404  
 405 104400  
 406  
 407 104000  
 408  
 409  
 410  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426

HLT=TRAP  
 SCOPE=EMT

;HLT IS A TRAP TO THE ERROR ROUTINE

;SCOPE IS AN EMT TRAP

;INDEX OF MACROS

.SCOPE  
 .SAVE  
 .REST  
 .ERROR  
 .PRINT  
 .DUMP  
 .RAND  
 .READ  
 .PACK

;INDEX OF CALLS

SCOPE  
 SAVE  
 REST  
 HLT  
 PRINT

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

.....  
DUMP  
DUMPF  
SDUMP  
SDUMPF  
RAND  
READ  
PACK  
.....

```

          .LIST ME
          =200
000200 012707 002336      MOV      #START,PC      ;GO TO START OF TEST
          =1000
001000 000000      ICNT: 0      ;CONTAINS PASS COUNT
001002 000000      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.
001004 032737 040000 177570  SCOPE$: BIT      #B14,2#SWR      ;LOOP ON ERROR?
001012 001403      BEQ      2$      ;BRANCH IF NO
001014 005767 000220      TST      ERRFLG      ;IS THERE AN ERROR?
001020 001003      BNE      1$      ;BRANCH IF YES
001022 005067 000212 2$: CLR      ERRFLG      ;RESET ERROR CONDITION
001026 000002      RTI      ;EXIT
001030 016716 177746 1$: MOV      LAD,(SP)      ;MODIFY RETURN ADDRESS
001034 000002      RTI      ;EXIT
          ;ROUTINE TO SAVE REGISTERS ON THE STACK.
          ;CALLED BY SAVE MACRO
001036 012667 000020  SAVES$: MOV      (SP)+,1$      ;SAVE RETURN PC
001042 010546      MOV      R5,-(SP)
001044 010446      MOV      R4,-(SP)
001046 010346      MOV      R3,-(SP)
001050 010246      MOV      R2,-(SP)
001052 010146      MOV      R1,-(SP)
001054 010046      MOV      R0,-(SP)
001056 016707 000000      MOV      1$,PC      ;RETURN

```

# MO1

FRONT END  
DZRPCB.MO1

MACY11 27(732) 16-SEP-76 15:29 PAGE 12

```

483 001062 000000 1$: 0 ;CONTAINS RETURN ADDRESS
484 ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
485 ;CALLED BY REST MACRO
486 001064 012667 000020 REST$: MOV (SP)+,1$ ;SAVE RETURN PC
487 001070 012600 MOV (SP)+,R0
488 001072 012601 MOV (SP)+,R1
489 001074 012602 MOV (SP)+,R2
490 001076 012603 MOV (SP)+,R3
491 001100 012604 MOV (SP)+,R4
492 001102 012605 MOV (SP)+,R5
493 001104 016707 000000 MOV 1$,PC ;RETURN
494 001110 000000 1$: 0 ;CONTAINS RETURN ADDR
495 ;ERROR SERVICE ROUTINE CALLED BY HLT
496 ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
497 ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
498 001112 005737 177570 ERROR: TST @#SWR ;HALT ON ERROR?
499 001116 100001 BPL 3$ ;BRANCH IF NO
500 081120 000000 HALT
501 001122 032737 004000 177570 3$: BIT #B11,@#SWR ;RING THE BELL?
502 001130 001403 BEQ 1$ ;BRANCH IF NO
503 001132 004567 000144 JSR R5,PRNTF$ ;FORCE PRINT THE MESSAGE
504 001136 001250 BELL
505 001140 032737 020000 177570 1$: BIT #B13,@#SWR ;SKIP TYPEOUT?
506 001146 001022 BNE 2$ ;BRANCH IF YES
507 001150 004567 000110 JSR R5,PRINT$ ;PRINT MESSAGE
508 001154 001252 ERRPC
509 001156 011667 000062 MOV (6),HLTADS ;GET ERROR PC+2
510 001162 162767 000002 000054 SUB #2,HLTADS ;MODIFY
511 001170 117767 000050 000044 MOV @HLTADS,HLTCTS ;SAVE HLT ARGUMENT
512 001176 016767 000042 000356 MOV HLTADS,TTY
513 001204 004767 000134 JSR PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
514 001210 004767 003670 JSR PC,MSG ;GO TO USER ERROR ROUTINE
515 001214 005737 177570 2$: TST @#SWR ;HALT ON ERROR?
516 001220 100001 BPL 4$ ;BRANCH IF NO
517 001222 000000 HALT
518 001224 052767 100000 000006 4$: BIS #B15,ERRFLG ;SET ERROR FLAG
519 001232 005267 000010 INC ERRORS ;UPDATE ERROR COUNTER
520 001236 000002 RTI
521 001240 000000 ERRFLG: 0
522 001242 000000 HLTCTS: 0
523 001244 000000 HLTADS: 0 ;PC OF ERROR
524 001246 000000 ERRORS: 0 ;ERROR COUNT
525 001250 000007 BELL: .ASCIZ <7>
526 001252 005015 005015 041520 ERRPC: .ASCIZ <15><12><15><12>'PC= '
527 001260 020075 000 .EVEN
528 001264 ;THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
529 ;THE MESSAGE MUST TERMINATE IN 0
530 PRINT$: BIT #B13,@#SWR ;INHIBIT TYPEOUTS?
531 001264 032737 020000 177570 BEQ PRNTF$ ;BRANCH IF NO
532 001272 001403 ADD #2,R5 ;UPDATE RETURN ADDR
533 001274 062705 000002 RTS R5
534 001300 000205 PRNTF$: TSTB @#TPS ;WAIT FOR PRINTER TO FINISH
535 001302 105737 177564 BPL -4
536 001306 100375 MOV R5,-(SP)
537 001310 010546 ADD #2,(SP) ;ADJUST RETURN PC
538 001312 062716 000002

```

```

539 001316 011505          MOV      (R5),R5          ;GET MESSAGE ADDR
540 001320 105715          1$:     TSTB     (R5)          ;CHECK FOR TERMINATOR
541 001322 001002          BNE     2$
542 001324 012605          MOV     (SP)+,R5          ;GET RETURN ADDR
543 001326 000205          RTS     R5                ;RETURN
544 001330 112537 177566    2$:     MOVB    (R5)+,@#TPB    ;PRINT CHARACTER
545 001334 105737 177564    TSTB   @#TPS             ;WAIT TILL DONE
546 001340 100375          BPL     -4
547 001342 000766          BR     1$
548                                     ;THIS ROUTINE TYPES A LOCATION IN OCTAL
549 001344 032737 020000 177570 PRINTR: BIT   #B13,@#SWR    ;INHIBIT TYPEOUT?
550 001352 001406          BEQ     PRINTA           ;BRANCH IF NO
551 001354 000207          RTS     PC
552 001356 032737 020000 177570 PRINTS: BIT   #B13,@#SWR    ;INHIBIT TYPEOUT?
553 001364 001405          BEQ     PRINTB           ;BRANCH IF NO
554 001366 000207          RTS     PC
555 001370 112767 000001 000140 PRINTA: MOVB  #1,.PR          ;SET ZERO FILL SWITCH
556 001376 000402          BR     .+6              ;SKIP
557 001400 005067 000132          PRINTB: CLR   .PR          ;SUPPRESS LEADING ZEROS
558 001404 112767 177772 000125 .PTIT:  MOV   #-6,.PR+1     ;SET COUNT
559 001412 010446          .PTIT:  MOV   R4,-(SP)     ;SAVE R4
560 001414 012704 001540          MOV     #.PR+2,R4       ;SET POINTER TO FIRST CHARACTER
561 001420 105014          CLRB   (R4)            ;CLEAR FIRST BYTE
562 001422 000413          BR     .PRF            ;ROTATE FIRST BIT
563 001424 105014          .PRL:  CLRB   (R4)       ;CLEAR BYTE OF CHAR
564 001426 032767 000100 000102 BIT     #100,.PR        ;BIT TYPING MODE
565 001434 001006          BNE    .PRF            ;YES SKIP 2 ROTATES
566 001436 006167 000120          ROL    TTY            ;ROTATE BIT INTO C
567 001442 106114          ROLB   (4)            ;PACK IT
568 001444 006167 000112          ROL    TTY
569 001450 106114          ROLB   (4)
570 001452 006167 000104          .PRF:  ROL    TTY
571 001456 106114          ROLB   (4)
572 001460 105714          TSTB   (4)            ;IS IT ZERO
573 001462 001402          BEQ    .+6            ;SKIP INC
574 001464 105267 000046          INCB   .PR            ;SET FILL SWITCH
575 001470 105767 000042          TSTB   .PR            ;CHECK FILL SWITCH
576 001474 001402          BEQ    .+6            ;SKIP BITSET
577 001476 152724 000060          BISB   #'0,(4)+       ;MAKE INTO ASCIZ CHAR
578 001502 105267 000031          INCB   .PR+1          ;INC COUNT
579 001506 001346          BNE    .PRL           ;REPEAT
580 001510 022704 001540          CMP    #.PR+2,R4      ;EMPTY BUFFER
581 001514 001002          BNE    .+6            ;SKIP IF NOT
582 001516 112724 000060          MOVB   #'0,(4)+       ;LOAD ONE ZERO
583 001522 105014          CLRB   (4)            ;NULL TERMINATOR
584 001524 004567 177534          JSR    R5,PRINTS      ;PRINT MESSAGE
585 001530 001540          .PR+2
586 001532 012604          MOV    (SP)+,R4       ;RESTORE R4
587 001534 000207          RTS     PC
588 001536 000012          .PR:   .BLKW   12
589 001562 000000          TTY:   0
590 001564          RAND$:
591 001564 004767 177246          JSR    PC,SAVE$       ;SAVE THE REGISTERS
592 001570 016700 000106          MOV    LO$UM,R0       ;SET R0 WITH LOW
593 001574 016701 000100          MOV    HI$UM,R1       ;SET R1 WITH HIGH
594 001600 012703 177771          MOV    #-7,R3         ;SET SHIFT COUNT

```

```

591 001604 005002          CLR      R2
592 001606 006300          ASL     R0
593 001610 006101          ROL     R1
594 001612 006102          ROL     R2
595 001614 005203          INC     R3
600 001616 001373          BNE     1$
601 001620 066702 000056  ADD     LONUM,R2
602 001624 005501          ADC     R1
603 001626 066701 000046  ADD     HINUM,R1
604 001632 005502          ADC     R2
605 001634 062700 001057  ADD     #1057,R0
606 001640 005501          ADC     R1
607 001642 005502          ADC     R2
608 001644 062701 047401  ADD     #47401,R1
609 001650 005502          ADC     R2
610 001652 062702 000006  ADD     #6,R2
611 001656 060200          ADD     R2,R0
612 001660 005501          ADC     R1
613 001662 010067 000014  MOV     R0,LONUM
614 001666 010167 000006  MOV     R1,HINUM
615 001672 004767 177166  JSR     PC,REST$
616 001676 000207          RTS     PC
617
618 001700 000000          HINUM: 0
619 001702 000000          LONUM: 0
620 001704 010346          READS: MOV     R3,-(6)
621 001706 012703 002014  1$:  MOV     #INPUT$,R3
622 001712 022703 002034  2$:  CMP     #INPUT$+20,R3
623 001716 001412          BEQ     4$
624 001720 105737 177560  TSTB   @#177560
625 001724 100375          BPL     -4
626 001726 113713 177562  MOVB   @#177562,(3)
627 001732 142713 000200  BICB   #200,(3)
628 001736 122713 000177  CMPB   #177,(3)
629 001742 001004          BNE     3$
630 001744          4$:
631 001744 004567 177314  JSR     R5,PRINT$
632 001750 002054          READMS
633 001752 000755          BR      1$
634 001754 013737 177562 177566  3$:  MOV     @#TKB,@#TPB
635 001762 105737 177564  TSTB   @#TPB
636 001766 100375          BPL     -4
637 001770 122723 000015  CMPB   #15,(3)+
638 001774 001346          BNE     2$
639 001776 105063 177777  CLRB   -1(3)
640 002002 004567 177256  JSR     R5,PRINT$
641 002006 002060          READLS
642 002010 012603          MOV     (6)+,R3
643 002012 000207          RTS     PC
644
645 002014 000020          INPUTS: .BLKW 20
646 002054 006477 000012  READMS: .ASCIZ '?'(15)<12>
647 002060 000012  READLS: .ASCIZ <12>
648
649
650
;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER

```

651									
652	002062								
653	002062	004767	176750		PACKS:	JSR	PC,SAVES		;SAVE THE REGISTERS
654	002066	005067	000242			CLR	NUMS		
655	002072	005000				CLR	RO		
656	002074	105760	002014		2S:	TSTB	INPUTS(RO)		
657	002100	001402				SEQ	1S		
658	002102	005200				INC	RO		
659	002104	000773				BR	2S		
660	002106	005300			1S:	DEC	RO		
661	002110	004767	000166			JSR	PC,PACS		;GET OCTAL CHAR
662	002114	016767	000212	000212		MOV	PK\$,NUMS		;PACK FIRST CHAR
663	002122	004767	000154			JSR	PC,PACS		;GET OCTAL CHAR
664	002126	000241				CLC			
665	002130	006167	000176			ROL	PK\$		
666	002134	006167	000172			ROL	PK\$		
667	002140	006167	000166			ROL	PK\$		
668	002144	056767	000162	000162		BIS	PK\$,NUMS		;PACK SECOND CHAR
669	002152	004767	000124			JSR	PC,PACS		;GET OCTAL CHAR
670	002156	000241				CLC			
671	002160	000367	000146			SWAB	PK\$		
672	002164	006067	000142			ROR	PK\$		
673	002170	006067	000136			ROR	PK\$		
674	002174	056767	000132	000132		BIS	PK\$,NUMS		;PACK THIRD CHAR
675	002202	004767	000074			JSR	PC,PACS		;GET OCTAL CHAR
676	002206	000367	000120			SWAB	PK\$		
677	002212	000241				CLC			
678	002214	006167	000112			ROL	PK\$		
679	002220	056767	000106	000106		BIS	PK\$,NUMS		;PACK FOURTH CHAR
680	002226	004767	000050			JSR	PC,PACS		;GET OCTAL CHAR
681	002232	000367	000074			SWAB	PK\$		
682	002236	000241				CLC			
683	002240	006167	000066			ROL	PK\$		
684	002244	006167	000062			ROL	PK\$		
685	002250	006167	000056			ROL	PK\$		
686	002254	006167	000052			ROL	PK\$		
687	002260	056767	000046	000046		BIS	PK\$,NUMS		;PACK FIFTH CHAR
688	002266	000402				BR	PKEX1\$		
689	002270	062706	000002		PKEX\$:	ADD	#2,SP		;MODIFY STACK
690	002274				PKEX1\$:				
691	002274	004767	176564			JSR	PC,RESTS		;RESTORE THE REGISTERS
692	002300	000207				RTS	PC		;EXIT
693									
694	002302	005700			PACS:	TST	RO		
695	002304	100771				BMI	PKEX\$		
696	002306	005067	000020			CLR	PK\$		
697	002312	116067	002014	000012		MOVB	INPUTS(RO),PK\$		;GET INPUT CHAR
698	002320	005300				DEC	RO		
699	002322	042767	177770	000002		BIC	#177770,PK\$		;CLEAR UNWANTED BITS
700	002330	000207				RTS	PC		
701									
702	002332	000000			PK\$:	0			
703	002334	000000			NUMS:	0			



002

FRONT END  
DZRPCB.P11

MAY11 27(732) 16-SEP-76 15:29 PAGE 17

704

```

705          .TITLE  RP11C MULTI DRIVE DIAGNOSTIC
706
707          002336  012706  000500          START:  MOV      #STKPTR,SP      ;SET STACK POINTER
708          002342  012737  000340  177776  MOV      #340,2#PSW      ;LOCK UP INTERRUPTS
709          002350  012767  001112  175456  MOV      #ERROR,34      ;SETUP ERROR TRAP
710          002356  012767  000340  175452  MOV      #PRI7,36
711          002364  012767  001004  175436  MOV      #SCOPE$,30      ;SETUP SCOPE TRAP
712          002372  012767  000340  175432  MOV      #PRI7,32
713          002400  012737  004732  000254  MOV      #DSKINT,2#VECTOR ;SET UP DISK INTERRUPT VECTOR
714          002406  012737  000340  000256  MOV      #340,2#STATUS
715          002414  005000
716          002416  005060  005374          CLRTAB: CLR      DEVTBL(RO)      ;CLEAR THE DEVICE TABLE
717          002422  005720          TST      (RO)+
718          002424  022700  000200          CMP      #128.,RO
719          002430  001372          BNE      CLRTAB
720          002432  005737  000042          TST      2#42          ;UNDER MONITOR CONTROL?
721          002436  001424          BEQ      5$          ;BRANCH IF NO
722          002440  005000          CLR      RO          ;CLEAR MODIFIER
723          002442  005001
724          002444  012777  000001  003146  7$:  MOV      #1,2#RPCS      ;CLEAR THE CONTROLLER
725          002452  110177  003144          MOVB     R1,2#RPCS1    ;SELECT UNIT
726          002456  005777  003154          TST      2#RPS        ;IS UNIT READY?
727          002462  100403          BMI      6$          ;BRANCH IF YES
728          002464  052760  100000  005374  BIS      #B15,DEVTBL(RO) ;SET UNIT UNAVAILABLE BIT
729          002472  062700  000020  6$:  ADD      #16.,RO      ;UPDATE MODIFIER
730          002476  005201          INC      R1          ;UPDATE UNIT NUMBER
731          002500  032701  000010          BIT      #B3,R1      ;ALL UNITS TESTED?
732          002504  001757          BEQ      7$          ;BRANCH IF NO
733          002506  000420
734          002510  012701  000001  5$:  MOV      #1,R1
735          002514  005000          CLR      RO
736          002516  030137  177570  2$:  BIT      R1,2#SWR      ;TEST THE SWITCH REGISTER
737          002522  001003          BNE      1$          ;TO DETERMINE WHICH UNITS
738          002524  052760  100000  005374  BIS      #B15,DEVTBL(RO) ;TO TEST. IF THE UNIT IS UNAVAILABLE
739          002532  062700  000020  1$:  ADD      #16.,RO      ;SET BIT 15 IN THE DEVICE TABLE
740          002536  000241
741          002540  006101
742          002542  032701  000400          ROL      R1
743          002546  001763          BIT      #B8,R1      ;HAVE ALL UNITS BEEN SCANNED?
744          002550  000005  8$:  BEQ      2$          ;NO-BRANCH
745          002552  004567  001712          RESET
746          002556  005067  003012          JSR      R5,EXTMEN    ;CLEAR THE SYSTEM
747          002562  005067  003010          CLR      UNIT        ;DETERMINE AMOUNT OF CORE
748          002566  005005          CLR      PASSCT      ;INITIALIZE POINTER
749          002570  032765  100000  005374  4$:  CLR      R5
750          002576  001002          BIT      #B15,DEVTBL(R5) ;IS UNIT AVAILABLE?
751          002600  004767  000136          BNE      3$          ;BRANCH IF NO
752          002604  005267  002764          JSR      PC,HOME     ;DO A HOME SEEK
753          002610  062705  000020  3$:  INC      UNIT        ;UPDATE UNIT
754          002614  032767  000010  002752  ADD      #16.,R5      ;UPDATE TABLE POINTER
755          002622  001762          BIT      #B3,UNIT     ;HAVE ALL UNITS BEEN HOMED?
756          002624  005067  002744          BEQ      4$          ;NO-BRANCH
757          002630  005005          LOOP:  CLR      UNIT
758          002632  032765  100000  005374  MAIN: CLR      R5
759          002640  001004          BIT      #B15,DEVTBL(R5) ;IS THE UNIT AVAILABLE?
760          002642  016504  005374          BNE      1$          ;BRANCH IF NO
          MOV      DEVTBL(R5),R4

```

```

761 002646 004774 003106      JSR      PC, JMPTBL(R4)  ;PERFORM FUNCTION IN JMPTBL
762 002652 005267 002716      1$:     INC      UNIT      ;UPDATE UNIT
763 002656 062705 000020      ADD     #15, R5         ;UPDATE TABLE POINTER
764 002662 032767 000010 002704  BIT     #B3, UNIT      ;HAVE ALL UNITS BEEN SCANNED?
765 002670 001760          BEQ     MAIN          ;NO BRANCH
766 002672 005267 002700      INC     PASSCT        ;INCREMENT ITERATION COUNTER
767 002676 016737 002674 177570  MOV     PASSCT, #SWR   ;DISPLAY COUNT
768 002704 005737 000042      TST     #42          ;UNDER MONITOR CONTROL?
769 002710 001413          BEQ     MEXIT1        ;BRANCH IF NO
770 002712 022767 005000 002656  CMP     #5000, PASSCT ;IS PASS COMPLETE?
771 002720 001007          BNE     MEXIT1        ;BRANCH IF NO
772 002722 013701 000042      MOV     #42, R1       ;GET RETURN ADDRESS
773 002726 000005          RESET
774 002730 004711      MEXIT: JSR      PC, (R1)  ;RETURN TO MONITOR
775 002732 000240      NOP
776 002734 000240      NOP
777 002736 000240      NOP
778 002740 000731      MEXIT1: BR     LOOP    ;LOOP
779
780      ;THIS ROUTINE WILL SEEK HOME THE PACK WHOSE
781      ;ADDRESS IS IN UNIT.
782
783 002742 116777 002626 002652  HOME:  MOVB   UNIT, #RPCS1  ;LOAD THE UNIT #
784 002750 005777 002662          TST     #RPCS        ;IS THE UNIT READY?
785 002754 100401          BMI     5$           ;BRANCH IF READY
786 002756 104400          HLT
787 002760 112777 000015 002632  5$:     MOVB   #15, #RPCS   ;UNIT IS NOT READY
788 002766 012704 000025          MOV     #25, R4     ;DO A HOME SEEK
789 002772 005304          1$:     DEC     R4         ;WAIT FOR SEEK TO START
790 002774 001376          BNE     1$
791 002776 032777 002000 002632  BIT     #B10, #RPCS  ;IS SEEK UNDER WAY SET
792 003004 001001          BNE     2$           ;YES
793 003006 104400          HLT                 ;SEEK UNDERWAY DID NOT SET
794 003010 016704 002560  2$:     MOV     UNIT, R4
795 003014 005067 000054          CLR     ATTNB
796 003020 116467 003076 000046  MOVB   ATTN(R4), ATTNB ;DETERMINE ATTENTION RESPONSE
797 003026 005000          CLR     RO
798 003030 005200  7$:     INC     RO
799 003032 001376          BNE     7$
800 003034 005200  6$:     INC     RO
801 003036 036777 000032 002572  BIT     ATTNB, #RPCS ;TIME OUT ATTENTION BIT
802 003044 001003          BNE     3$           ;DID ATTENTION SET?
803 003046 005700          TST     RO
804 003050 001371          BNE     6$           ;BRANCH IF YES
805 003052 104400          HLT                 ;DID IT TIME OUT?
806      ;ATTENTION BIT DID NOT SET
807 003054 005777 002540  3$:     TST     #RPCS        ;ANY DEVICE STATUS ERRORS?
808 003060 100001          BPL     4$           ;NO-BRANCH
809 003062 104400          HLT
810 003064 116777 000004 002544  4$:     MOVB   ATTNB, #RPCS ;DEVICE STATUS ERROR AFTER HOME COMMAND
811 003072 000207          RTS     PC          ;CLEAR ATTENTION BIT
812
813 003074 000000      ATTNB: 0             ;CONTAINS ATTENTION BIT FOR CURRENT UNIT
814 003076 001 002 004  ATTN:  .BYTE 1,2,4,10,20,40,100,200
815 003101 010 020 040
816 003104 100 200

```

817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872

003106	003116			
003110	003432			
003112	003744			
003114	004120			
016565	005400	005376		
116777	002444	002470		
004767	176426			
016767	176540	002436		
016767	176530	002432		
032777	020000	002456		
001410				
042767	177000	002412		
022767	000625	002404		
002755				
000407				
042767	177400	002372		
022767	000312	002364		
002745				
026765	002356	005376		
001741				
016765	002346	005400		
016765	002342	005402		
004767	176314			
016767	176426	002324		
016767	176416	002320		
042767	177760	002310		
022767	000011	002302		
002003				
162767	000010	002272		
042767	177740	002266		
022767	000023	002260		
002003				
162767	000014	002250		
116767	002244	002241		
016765	002234	005404		
005065	005406			
016577	005400	002246		
016577	005404	002242		
112777	000011	002222		
012704	000025			
005304				

```

      .EVEN
; THIS TABLE DETERMINES WHERE CONTROL WILL GO FOR ANY
; PARTICULAR UNIT. THE FIRST WORD OF THE DEVICE TABLE
; IS USED AS A MODIFIER FOR A JSR INDIRECT INTO
; THE FOLLOWING TABLE.

JMPTBL: SEEK          ;SEEK A RANDOM CYLINDER
          SEKCK        ;SEE IF SEEK IS COMPLETE
          WRITE        ;WRITE RANDOM DATA
          READD        ;READ AND VERIFY RANDOM DATA

; THIS ROUTINE WILL GENERATE A RANDOM CYLINDER
; ADDRESS AND ISSUE A SEEK TO IT. THE FUNCTION
; POINTER IN THE DEVICE TABLE WILL BE UPDATED TO
; CHECK FOR THE ATTENTION BIT.

SEEK:    MOV          DEVTBL+4(R5),DEVTBL+2(R5) ;SET UP CYLINDER FROM ADDRESS
          MOVB        UNIT,ARPCSI             ;SELECT THE UNIT
1$:      JSR          PC,RANDS                ;GENERATE TWO RANDOM NOS.
          MOV          LONUM,WORK1
          MOV          HINUM,WORK2
          BIT          #813,ARPCSI           ;IS THIS AN RPO3?
          BEQ          6$                    ;BRANCH IF NO
          BIC          #177000,WORK1
          CMP          #625,WORK1           ;GENERATE A CYLINDER ADDRESS
          BLT          1$
          BR          7$
6$:      BIC          #177400,WORK1
          CMP          #312,WORK1           ;IS NUMBER TOO LARGE?
          BLT          1$                   ;BRANCH IF YES
7$:      CMP          WORK1,DEVTBL+2(R5)
          BEQ          1$
          MOV          WORK1,DEVTBL+4(R5)    ;SAVE CYLINDER ADDRESS
          MOV          WORK2,DEVTBL+6(R5)    ;USE A RANDOM DATA BASE
          JSR          PC,RANDS              ;GENERATE TWO RANDOM NOS.
          MOV          LONUM,WORK1
          MOV          HINUM,WORK2
          BIC          #177760,WORK1
          CMP          #11,WORK1            ;GENERATE A RANDOM SECTOR
          BGE          2$
          SUB          #8,WORK1
          BIC          #177740,WORK2        ;GENERATE A RANDOM TRACK
          CMP          #23,WORK2
          BGE          3$
          SUB          #14,WORK2
          MOVB        WORK2,WORK1+1         ;MERGE TRACK AND SECTOR ADDR
3$:      MOV          WORK1,DEVTBL+10(R5)    ;STORE RANDOM DISK ADDRESS
          CLR          DEVTBL+12(R5)        ;CLEAR TIMEOUT COUNTER
          MOV          DEVTBL+4(R5),ARPCA    ;LOAD CYLINDER ADDRESS
          MOV          DEVTBL+10(R5),ARPCDA  ;LOAD TRACK AND SECTOR
          MOVB        #11,ARPCSI           ;INITIATE A SEEK
          MOV          #25,R4
4$:      DEC          R4                    ;WAIT FOR SEEK TO START
    
```

```

873 003404 001376          BNE      4$
874 003406 032777 002000 002222  BIT      #B10,ARPDS      ;IS THE SEEK UNDERWAY?
875 003414 001001          BNE      5$              ;YES-BRANCH
876 003416 104400          HLT
877 003420 005265 005374 5$:  INC      DEVTBL(R5)      ;SEEK UNDERWAY DID NOT SET
878 003424 005265 005374  INC      DEVTBL(R5)      ;MODIFY FUNCTION POINTER TO
879 003430 000207          RTS      PC              ;CHECK FOR SEEK COMPLETE
880                                     ;EXIT
881                                     ;THIS ROUTINE IS ENTERED AFTER A SEEK HAS BEEN ISSUED.
882                                     ;IT CHECKS THE ATTENTION FLAG - IF CLEAR IT UPDATES THE
883                                     ;TIMEOUT COUNTER AND CHECKS IT. IF SET IT VERIFIES
884                                     ;THE SEEK FUNCTIONED PROPERLY.
885
886 003432 016704 002136  SEKCK:  MOV      UNIT,R4
887 003436 116467 003076 177430  MOVB     ATTN(R4),ATTNB ;DETERMINE ATTENTION BIT
888 003444 036777 177424 002164  BIT      ATTNB,ARPDS    ;TEST FOR ATTENTION FLAG
889 003452 001014          BNE      1$              ;BRANCH IF SET
890 003454 005265 005406          INC      DEVTBL+12(R5)   ;UPDATE TIMEOUT COUNTER
891 003460 022765 005000 005406  CMP      #5000,DEVTBL+12(R5) ;DID OPERATION TIMEOUT?
892 003466 101005          BHI      2$              ;NOT YET-BRANCH
893 003470 116777 002100 002124  MOVB     UNIT,ARPCSI    ;SELECT UNIT
894 003476 104400          HLT
895 003500 000447          BR       4$              ;SEEK TIMED OUT
896 003502 000207          RTS      PC              ;EXIT
897 003504 116777 177364 002124 2$:  MOVB     ATTNB,ARPDS    ;CLEAR ATTENTION FLAG
898 003512 116777 002056 002102 1$:  MOVB     UNIT,ARPCSI    ;SELECT UNIT
899 003520 032777 002000 002110  BIT      #B10,ARPDS    ;IS SEEK UNDERWAY CLEAR?
900 003526 001402          BEQ      3$              ;IF YES-BRANCH
901 003530 104400          HLT
902 003532 000432          BR       4$              ;SEEK UNDERWAY DID NOT CLEAR
903 003534 005777 002060 3$:  TST      ARPCS
904 003540 100002          BPL      5$              ;ARE THERE ANY DEVICE STATUS ERRORS?
905 003542 104400          HLT
906 003544 000425          BR       4$              ;BRANCH-NO ERRORS
907 003546 017704 002066 5$:  MOV      JSUCA,R4        ;DEVICE STATUS ERRORS
908 003552 020465 005400          CMP      R4,DEVTBL+4(R5) ;GET CURRENT CYLINDER ADDRESS
909 003556 001440          BEQ      6$              ;DOES IT MATCH CYLINDER REQUESTED?
910 003560 104400          HLT
911 003562 004567 175476          JSR      R5,PRINT$      ;YES-BRANCH
912 003566 003672          MES10
913 003570 016567 005400 175764  MOV      DEVTBL+4(R5),TTY ;SUCA AND CYL REQUESTED DID NOT COMPARE
914 003576 004767 175554          JSR      PC,PRINT$      ;PRINT MESSAGE
915 003602 004567 175456          JSR      R5,PRINT$      ;TYPE LOCATION-SUPRESS ZEROS
916 003606 003721          MES11
917 003610 010467 175746          MOV      R4,TTY
918 003614 004767 175524          JSR      PC,PRINT$      ;PRINT MESSAGE
919 003620 032777 004000 002010 4$:  BIT      #B11,ARPDS    ;TYPE LOCATION WITH LEADING ZEROS
920 003626 001411          BEQ      10$           ;SEEK INCOMPLETE?
921 003630 112777 000015 001762  MOVB     #15,ARPCS      ;BRANCH IF NO
922 003636 105777 001756          TSTB     ARPCS          ;ISSUE HOME COMMAND
923 003642 100375          BPL      -4             ;WAIT FOR DONE
924 003644 005777 001766          TST      ARPDS         ;WAIT FOR UNIT READY
925 003650 100375          BPL      -4
926 003652 005065 005374 10$:  CLR      DEVTBL(R5)     ;CLEAR FUNCTION POINTER
927 003656 000207          RTS      PC              ;EXIT
928 003660 005265 005374 6$:  INC      DEVTBL(R5)

```

```

929 003664 005265 005374      INC      DEVTBL(R5)      ;UPDATE FUNCTION POINTER
930 003670 000207              RTS      PC              ;EXIT
931
932 003672 005015 042522 052521 MES10: .ASCIZ <15><12>'REQUESTED CYLINDER= '
933 003700 051505 042524 020104
934 003706 054503 044514 042116
935 003714 051105 020075      000
936 003721      015 051412 041525 MES11: .ASCIZ <15><12>'SUCA REGISTER= '
937 003726 020101 042522 044507
938 003734 052123 051105 020075
939 003742      000
940      003744              .EVEN
941
942      ;THIS ROUTINE WILL WRITE A RANDOM SECTOR ON:
943      ;A RANDOM TRACK. THE CYLINDER HAS ALREADY
944      ;BEEN SELEYCED BY THE SEEK ROUTINE.
945
946 003744 004767 000636      WRITE: JSR      PC,RANADR      ;GENERATE A RANDOM BUFFER ADDR
947 003750 012767 000400 001622      MOV      #400,WORK
948 003756 016701 001630      MOV      BUFF,R1
949 003762 004767 000336      JSR      PC,RANDAT      ;GENERATE A RANDOM PATTERN
950 003766 116777 001602 001626      MOVVB   UNIT,ARPCSI      ;SELECT THE UNIT
951 003774 032777 100000 001634      BIT      #B15,ARPDS      ;IS THE SELECTED UNIT READY
952 004002 001003              BNE      1$              ;YES-BRANCH
953 004004 104400              HLT
954 004006 000167 000100      JMP      WRTER          ;SELECTED UNIT NOT READY
955                          ;START SEQUENCE OVER
956 004012 012777 177400 001604 1$:      MOV      #-400,ARPWC      ;SETUP WORD COUNT REGISTER
957 004020 016777 001566 001600      MOV      BUFF,ARPBA      ;SETUP BUS ADDR REGISTER
958 004026 016577 005400 001574      MOV      DEVTBL+4(R5),ARPCA ;SET CYLINDER ADDR
959 004034 016577 005404 001570      MOV      DEVTBL+10(R5),ARPDA ;SETUP DISK ADDR.
960 004042 005067 001546      CLR      INT              ;CLEAR INTERRUPT FLAG
961 004046 012737 000200 177776      MOV      #PRI4,ARPSW      ;ALLOW INTERRUPT
962 004054 005067 001530      CLR      INTERR          ;CLEAR ERROR FLAG
963 004060 112777 000113 001532      MOVVB   #113,ARPCS      ;INITIATE WRITE WITH INTERRUPT
964 004066 004767 000612      JSR      PC,WAIT          ;TIMEOUT THE OPERATION
965 004072 005767 001512      TST      INTERR          ;ANY ERRORS?
966 004076 001005      BNE      WRTER          ;BRANCH IF YES
967 004100 005265 005374      INC      DEVTBL(R5)      ;UPDATE FUNCTION POINTER TO READ
968 004104 005265 005374      INC      DEVTBL(R5)
969 004110 000403      BR       READD
970 004112 005065 005374      WRTER: CLR      DEVTBL(R5) ;RESTORE FUNCTION POINTER
971 004116 000207      RTS      PC              ;EXIT
972
973      ;READ AND VERIFY THE DATA WRITTEN
974
975 004120 116777 001450 001474 READD: MOVVB   UNIT,ARPCSI      ;SELECT THE UNIT
976 004126 032777 100000 001502      BIT      #B15,ARPDS      ;IS THE SELECTED UNIT READY?
977 004134 001003              BNE      1$              ;YES-BRANCH
978 004136 104400              HLT          ;SELECTED UNIT NOT READY
979 004140 000167 000152      JMP      RDCNT
980 004144 012777 177400 001452 1$:      MOV      #-400,ARPWC      ;LOAD WORD COUNT REGISTER
981 004152 016777 001434 001446      MOV      BUFF,ARPBA      ;LOAD BUS ADDR REGISTER
982 004160 062777 001000 001440      ADD      #1000,ARPBA
983 004166 016577 005400 001434      MOV      DEVTBL+4(R5),ARPCA ;SET CYLINDER ADDR
984 004174 016577 005404 001430      MOV      DEVTBL+10(R5),ARPDA ;SETUP DISK ADDR.

```

```

985 004202 005067 001406 CLR INT ;CLEAR INTERRUPT FLAG
986 004206 005067 001376 CLR INTERR ;CLEAR ERROR FLAG
987 004212 112777 000117 001400 MOVB #117,DRPCS ;INITIATE READ WITH INTERRUPT
988 004220 004767 000460 JSR PC,WAIT ;TIMEOUT THE OPERATION
989 004224 032777 040000 001366 BIT #814,DRPCS ;ANY HARD ERRORS?
990 004232 001031 BNE RDCNT ;BRANCH IF YES
991 004234 016701 001352 MOV BUFF,R1
992 004240 016702 001346 MOV BUFF,R2
993 004244 005003 CLR R3
994 004246 062701 001000 ADD #1000,R1 ;START OF PATTERN BUFFER
995 004252 022122 3$: CMP (R1)+,(R2)+ ;COMPARE DATA
996 004254 001006 BNE 2$ ;BRANCH-DATA DID NOT COMPARE
997 004256 005203 INC R3 ;INCREMENT COUNTER
998 004260 022703 000400 CMP #400,R3 ;HAS BUFFER BEEN SCANNED
999 004264 001372 BNE 3$ ;BRANCH-NO
1000 004266 000167 000024 JMP RDCNT
1001
1002 004272 016267 177776 001070 2$: MOV -2(R2),EXPS
1003 004300 016167 177776 001064 MOV -2(R1),RECS
1004 004306 004567 174752 JSR R5,PRINT$ ;PRINT MESSAGE
1005 004312 005747 MES12
1006 004314 104401 HLT +1 ;DATA DID NOT VERIFY
1007 004316 005065 005374 RDCNT: CLR DEVTBL(R5) ;INITIATE FUNCTION POINTER
1008 004322 000207 RTS PC ;EXIT
1009
1010 ;GENERATE A RANDOM PATTERN
1011
1012 004324 016567 005402 000132 RANDAT: MOV DEVTBL+6(R5),RAND1 ;GET RANDOM BASE
1013 004332 016567 005404 000126 MOV DEVTBL+10(R5),RAND2
1014 004340 016700 000120 MOV RAND1,R0
1015 004344 016704 000116 MOV RAND2,R4
1016 004350 012703 000007 RANDA1: MOV #7,R3 ;SETUP SHIFT COUNT
1017 004354 005002 CLR R2
1018 004356 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
1019 004360 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R0 INTO R4
1020 004362 006102 ROL R2 ;ROTATE CARRY OUT OF R4 INTO R2
1021 004364 005303 DEC R3 ;DECREMENT R3
1022 004366 001373 BNE SHIFT ;CONTINUE LOOP
1023 004370 066700 000070 ADD RAND1,R0 ;ADD IN # TO MAKE X129
1024 004374 005504 ADC R4 ;PROPOGATE CARRY
1025 004376 066704 000064 ADD RAND2,R4 ;ADD IN # TO MAKE X129
1026 004402 005502 ADC R2 ;PROPOGATE CARRY
1027 004404 062700 001057 ADD #1057,R0 ;ADD LOW CONSTANT
1028 004410 005504 ADC R4 ;PROPOGATE CARRY
1029 004412 005502 ADC R2 ;PROPOGATE AGAIN
1030 004414 062704 047401 ADD #47401,R4 ;ADD HIGH CONSTANT
1031 004420 005502 ADC R2
1032 004422 062702 000006 ADD #6,R2
1033 004426 060200 ADD R2,R0 ;REPRIME R0 WITH HIGH DIGIT
1034 004430 005504 ADC R4
1035 004432 010067 000026 MOV R0,RAND1
1036 004436 010021 MOV R0,(R1)+ ;STORE DATA IN BUFFER
1037 004440 005367 001134 DEC WORK
1038 004444 001406 BEQ EXGEN
1039 004446 010467 000014 MOV R4,RAND2
1040 004452 010421 MOV R4,(R1)+ ;STORE DATA IN BUFFER

```

```

1041 004454 005367 001120          DEC      WORK
1042 004460 001333                   BNE     RANDA1      ;FILL ENTIRE BUFFER
1043 004462 000207          EXGEN:  RTS      PC      ;EXIT
1044
1045 004464 000000          RAND1:  0
1046 004466 000000          RAND2:  0
1047
1048          ;THIS ROUTINE DETERMINES THE TOTAL AMOUNT OF AVAILABLE
1049          ;CARE WITHOUT USING MEMORY MANAGEMENT.
1050
1051 004470 012737 000340 177776  EXTMEN: MOV      #PRI7, @#PSW      ;LOCKUP PRIORITY LEVEL
1052 004476 012767 004546 173300          MOV      #MAXREF, 4      ;SETUP IO BUS TRAP
1053 004504 012767 000340 173274          MOV      #PRI7, 6
1054 004512 012767 017446 000064          MOV      #17446, SAVE      ;START WITH 4K
1055 004520 005777 000060          EXREF:  TST     @SAVE      ;REFERENCE MEMORY
1056 004524 022767 157446 000052          CMP      #157446, SAVE      ;TEST FOR 28K
1057 004532 001001          BNE     1$      ;BRANCH IF LESS THAN 28K
1058 004534 000407          BR      MAXRF1
1059 004536 062767 020000 000040  1$:    ADD      #20000, SAVE      ;SETUP FOR NEXT REFERENCE
1060 004544 000765          BR      EXREF
1061
1062          ;ENTER HERE WHEN IO BUS ERROR OCCURS
1063
1064 004546 162767 020000 000030  MAXREF: SUB      #20000, SAVE
1065 004554 012767 000006 173222  MAXRF1: MOV      #6, 4      ;RESTORE IO BUS TRAY
1066 004562 005067 173220          CLR      6
1067 004566 005737 000042          TST     @#42      ;UNDER MONITOR CONTROL?
1068 004572 001403          BEQ     1$      ;BRANCH IF NO
1069 004574 162767 005670 000002          SUB      #3000., SAVE      ;ADJUST TOP OF CORE
1070 004602 000205          1$:    RTS      R5      ;EXIT-SAVE=MAXIMUM MEMORY
1071 004604 000000          SAVE:  0      ;HIGHEST AVAILABLE LOCATION
1072
1073          ;GENERATE A RANDOM BUFFER ADDRESS
1074
1075 004606 016704 177772          RANADR: MOV      SAV, R4
1076 004612 162704 006054          SUB      #ENDP, R4      ;DETERMINE BUFFER SIZE
1077 004616 162704 002000          SUB      #2000, R4      ;ALLOW ROOM FOR DATA
1078 004622 004767 174736          JSR     PC, RAND$      ;GENERATE TWO RANDOM NOS.
1079 004626 016767 175050 000746          MOV      LONUM, WORK1
1080 004634 042767 000001 000740          BIC     #80, WORK1      ;MAKE NUMBER EVEN
1081 004642 012703 100000          MOV      #100000, R3
1082 004646 020467 000730          2$:    CMP      R4, WORK1      ;ENSURE THAT THE RANDOM
1083 004652 101005          BHI     1$      ;ADDRESS FITS WITHIN AVAILABLE
1084 004654 040367 000722          BIC     R3, WORK1      ;MEMORY
1085 004660 000241          CLC
1086 004662 006003          ROR
1087 004664 000770          BR      R3
1088 004666 062767 006054 000706  1$:    ADD      #ENDP, WORK1
1089 004674 016767 000702 000710          MOV      WORK1, BUFF      ;SAVE BUFFER START ADDR.
1090 004702 000207          RTS      PC      ;EXIT
1091
1092          ;TIMEOUT THE OCCURANCE OF AN INTERRUPT
1093
1093 004704 005000          WAIT:  CLR      R0
1094 004706 005200          2$:    INC      R0
1095 004710 005767 000700          TST     INT      ;HAS INTERRUPT OCCURED?
1096 004714 001005          BNE     1$      ;YES-BRANCH

```



```

1097 004716 005700          TST      RO          ;HAS OPERATION TIMED OUT?
1098 004720 001372          BNE     2$          ;NO-BRANCH
1099 004722 104400          HLT                    ;UNIT TIMED OUT ON READ OR WRITE
1100 004724 005267 000660  INC     INTERR      ;SET ERROR FLAG
1101 004730 000207          RTS     PC          ;EXIT
1102
1103          ;ENTERED UPON A DEVICE INTERRUPT. THIS ROUTINE
1104          ;WILL CHECK FOR AND REPORT DEVICE ERRORS
1105
1106 004732 032777 100000 000660 DSKINT: BIT      #B15,DRPCS ;WHERE THER ANY ERRORS?
1107 004740 001402          BEQ     1$          ;BRANCH-NO ERRORS
1108 004742 000167 000110  JMP     DSKER      ;REPORT ERROR
1109 004746 016703 000640 1$: MOV     BUFF,R3
1110 004752 062703 001000  ADD     #1000,R3
1111 004756 022765 000004 005374  CMP     #4,DEVTBL(R5) ;IS THIS A WRITE?
1112 004764 001402          BEQ     3$          ;BRANCH IF YES
1113 004766 062703 001000  ADD     #1000,R3
1114 004772 020377 000630 3$: CMP     R3,DRPBA ;DID THE BUS ADDR TERMINATE PROPERLY?
1115 004776 001425          BEQ     2$          ;YES-BRANCH
1116 005000 104400          HLT                    ;CONTENTS OF RPBA INCORRECT
1117 005002 004567 174256  JSR     R5,PRINT$ ;PRINT MESSAGE
1118 005006 005774          MES13
1119 005010 004567 174250  JSR     R5,PRINT$ ;PRINT MESSAGE
1120 005014 006024          MES18
1121 005016 010367 174540  MOV     R3,TTY
1122 005022 004767 174316  JSR     PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
1123 005026 004567 174232  JSR     R5,PRINT$ ;PRINT MESSAGE
1124 005032 006040          MES19
1125 005034 017767 000566 174520  MOV     DRPBA,TTY
1126 005042 004767 174276  JSR     PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
1127 005046 005267 000536  INC     INTERR      ;SET ERROR FLAG
1128 005052 000167 000006 2$: JMP     EXTINT
1129 005056 104400          DSKER: HLT                    ;REPORT INTERRUPT DISK ERROR
1130 005060 005267 000524  INC     INTERR      ;SET ERROR FLAG
1131 005064 052767 000001 000522  EXTINT: BIS     #B0,INT
1132 005072 032777 100000 000536 1$: BIT     #B15,DRPDS ;IS THE UNIT READY
1133 005100 001774          BEQ     1$          ;NO-WAIT
1134 005102 000002          RTI                    ;EXIT INTERRUPT
1135
1136
1137 005104 032767 000002 174130  MSG:  BIT     #B1,HLTCT$ ;TYPE ENTIRE MSG?
1138 005112 001100          BNE     1$          ;BRANCH IF NO
1139 005114 004567 174144  JSR     R5,PRINT$ ;PRINT MESSAGE
1140 005120 005642          MES1
1141 005122 016767 000446 174432  MOV     UNIT,TTY
1142 005130 004767 174222  JSR     PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
1143 005134 004567 174124  JSR     R5,PRINT$ ;PRINT MESSAGE
1144 005140 005674          MES2A
1145 005142 017767 000470 174412  MOV     DRPDS,TTY
1146 005150 004767 174170  JSR     PC,PRINTR
1147 005154 004567 174104  JSR     R5,PRINT$ ;PRINT MESSAGE
1148 005160 005652          MES1A
1149 005162 017767 000446 174372  MOV     DRPER,TTY
1150 005170 004767 174150  JSR     PC,PRINTR ;TYPE LOCATION WITH LEADING ZEROS
1151 005174 004567 174064  JSR     R5,PRINT$ ;PRINT MESSAGE
1152 005200 005663          MES2

```

```

1153 005202 017767 000412 174352      MOV      @RPCS, TTY
1154 005210 004767 174130      JSR      PC, PRINTR      ;TYPE LOCATION WITH LEADING ZEROS
1155 005214 004567 174044      JSR      RS, PRINTS     ;PRINT MESSAGE
1156 005220 005705                      MES3
1157 005222 016567 005400 174332      MOV      DEVTBL+4(R5), TTY
1158 005230 004767 174122      JSR      PC, PRINTS     ;TYPE LOCATION-SUPRESS ZEROS
1159 005234 004567 174024      JSR      RS, PRINTS     ;PRINT MESSAGE
1160 005240 005722                      MES4
1161 005242 005067 000340      CLR      WORK3
1162 005246 116567 005405 000332      MOVB    DEVTBL+11(R5), WORK3
1163 005254 016767 000326 174300      MOV      WORK3, TTY
1164 005262 004767 174070      JSR      PC, PRINTS     ;TYPE LOCATION-SUPRESS ZEROS
1165 005266 004567 173772      JSR      RS, PRINTS     ;PRINT MESSAGE
1166 005272 005734                      MES5
1167 005274 116567 005404 000304      MOVB    DEVTBL+10(R5), WORK3
1168 005302 016767 000300 174252      MOV      WORK3, TTY
1169 005310 004767 174042      JSR      PC, PRINTS     ;TYPE LOCATION-SUPRESS ZEROS
1170 005314 032767 000001 173720 1$:      BIT      #80, HLTCTS    ;TYPE EXPECTED - RECEIVED?
1171 005322 001001                      BNE     2$              ;BRANCH IF YES
1172 005324 000207                      RTS      PC
1173 005326                      2$:
1174 005326 004567 173732      JSR      RS, PRINTS     ;PRINT MESSAGE
1175 005332 006024                      MES18
1176 005334 016767 000030 174220      MOV      EXPS, TTY
1177 005342 004767 173776      JSR      PC, PRINTR     ;TYPE LOCATION WITH LEADING ZEROS
1178 005346 004567 173712      JSR      RS, PRINTS     ;PRINT MESSAGE
1179 005352 006040                      MES19
1180 005354 016767 000012 174200      MOV      RECS, TTY
1181 005362 004767 173756      JSR      PC, PRINTR     ;TYPE LOCATION WITH LEADING ZEROS
1182 005366 000207                      RTS      PC
1183 005370 000000      EXPS:    0
1184 005372 000000      RECS:    0
1185                      ;DEVTBL IS A TABLE CONTAINING SLOTS FOR EACH OF EIGHT
1186                      ;POSSIBLE UNITS. DURING THE OPERATION OF THE PROGRAM
1187                      ;RS IS USED AS A MODIFIER TO POINT INTO THE TABLE TO
1188                      ;SELECT THE PROPER UNIT. EACH UNIT SLOT CONTAINS
1189                      ;EIGHT ENTRIES(WORD)
1190                      ;1 FUNCTION POINTER
1191                      ;   0=SEEK
1192                      ;   2=SEEK IN PROGRESS
1193                      ;   4=WRITE
1194                      ;   6=READ
1195                      ; IF NEGATIVE-UNIT IS NOT TESTED
1196                      ;2 CYLINDER FROM ADDRESS-INDICATES PREVIOUS CYLINDER POSITION
1197                      ;3 CYLINDER TO ADDRESS-ADDRESS OF THE SEEK COMMAND
1198                      ;4 RANDOM BASE FOR PATTERN GENERATION
1199                      ;5 RANDOM TRACK AND SECTOR ADDRESS
1200                      ;6 CYLINDER SEEK TIMEOUT COUNTER
1201                      ;7 SPARE
1202                      ;8 SPARE
1203
1204                      .EVEN
1205 005374 000000      DEVTBL: 0              ;UNIT 0 SLOT
1206                      .+DEVTBL+20
1207 005414 000000      UNIT1: 0              ;UNIT 1 SLOT
1208                      .+UNIT1+20

```

```

1209 005434 000000 UNIT2: 0 ;UNIT 2 SLOT
1210 005454 005454 . =UNIT2+20
1211 005454 000000 UNIT3: 0 ;UNIT 3 SLOT
1212 005474 005474 . =UNIT3+20
1213 005474 000000 UNIT4: 0 ;UNIT 4 SLOT
1214 005514 005514 . =UNIT4+20
1215 005514 000000 UNITS: 0 ;UNIT 5 SLOT
1216 005534 005534 . =UNITS+20
1217 005534 000000 UNIT6: 0 ;UNIT6 SLOT
1218 005554 005554 . =UNIT6+20
1219 005554 000000 UNIT7: 0 ;UNIT 7 SLOT
1220 005574 005574 . =UNIT7+20
1221
1222 ;RP11 CONSTANTS-MEMORY ASSIGNMENTS
1223 005574 000000 UNIT: 0 ;CURRENT UNIT UNDER TEST
1224 005576 000000 PASSCT: 0 ;COUNTS EACH PASS THRU 8 UNITS
1225 005600 000000 WORK: 0 ;TEMPORARY STORAGE AREA
1226 005602 000000 WORK1: 0
1227 005604 000000 WORK2: 0
1228 005606 000000 WORK3: 0
1229 005610 000000 INTERR: 0 ;INTERRUPT ERROR FLAG
1230 005612 000000 BUFF: 0 ;STARTING ADDRESS OF BUFFER
1231 005614 000000 INT: 0 ;INTERRUPT FLAG
1232 005616 000000 FLAG: 0 ;FLAG WORD
1233 ;DISK IO REGISTERS
1234 005620 176714 RPCS: 176714 ;DISK CONTROL REGISTER
1235 005622 176715 RPCS1: 176715
1236 005624 176716 RPWC: 176716 ;DISK WORD COUNT REGISTER
1237 005626 176720 RPBA: 176720 ;CURRENT ADDRESS REGISTER
1238 005630 176722 RPCA: 176722 ;CYLINDER ADDRESS REGISTER
1239 005632 176724 RPDA: 176724 ;DISK ADDRESS REGISTER
1240 005634 176712 RPER: 176712 ;ERROR REGISTER
1241 005636 176710 RPDS: 176710 ;DRIVE STATUS REGISTER
1242 005640 176734 SUCA: 176734 ;CURRENT CYLINDER ADDRESS
1243 000254 VECTOR=254 ;INTERRUPT VECTOR ADDR.
1244 000256 STATUS=256 ;DISK INTERRUPT STATUS
1245
1246 ;MESSAGES
1247
1248 005642 005015 047125 052111 MES1: .ASCIZ <15><12>/UNIT /
1249 005650 000040
1250 005652 005015 050122 051105 MES1A: .ASCIZ <15><12>/RPER= /
1251 005660 020075 000
1252 005663 015 051012 041520 MES2: .ASCIZ <15><12>/RPCS= /
1253 005670 036523 000040
1254 005674 005015 050122 051504 MES2A: .ASCIZ <15><12>/RPDS= /
1255 005702 020075 000
1256 005705 015 041412 046131 MES3: .ASCIZ <15><12>/CYLINDER= /
1257 005712 047111 042504 036522
1258 005720 000040
1259 005722 005015 051124 041501 MES4: .ASCIZ <15><12>/TRACK= /
1260 005730 036513 000011
1261 005734 005015 042523 052103 MES5: .ASCIZ <15><12>/SECTOR= /
1262 005742 051117 020075 000
1263 005747 015 042012 052101 MES12: .ASCIZ <15><12>/DATA COMPARE ERROR/
1264 005754 020101 047503 050115

```

1265	005762	051101	020105	051105	
1266	005770	047522	000122		
1267	005774	005015	052502	020123	MES13: .ASCIZ <15><12>/BUS ADDRESS INCORRECT/
1268	006002	042101	051104	051505	
1269	006010	020123	047111	047503	
1270	006016	051122	041505	000124	
1271	006024	005015	054105	042520	MES18: .ASCIZ <15><12>/EXPECTED /
1272	006032	052103	042105	000040	
1273	006040	005015	042522	042503	MES19: .ASCIZ <15><12>/RECEIVED /
1274	006046	053111	042105	000040	
1275	006054	000000			ENDP: 0 ;START OF BUFFER AREA
1276		000001			.END















ROR	672	673	1086												
RTI	470	472	520	1134											
RTS	534	543	551	554	587	616	643	692	700	811	879	896	927	930	971
	1008	1043	1070	1090	1101	1172	1182								
SUB	510	859	863	1064	1069	1076	1077								
SWAB	671	676	681												
TRAP	405														
TST	467	498	515	694	717	720	726	768	784	803	807	903	924	965	1055
	1067	1095	1097												
TSTB	535	540	545	572	575	624	635	656	922						
.ABS	312														
.ASCIZ	525	526	646	647	932	936	1248	1250	1252	1254	1256	1259	1261	1263	1267
	1271	1273													
.BLKW	588	645													
.BYTE	814														
.END	1276														
.ENDC	515														
.EVEN	528	812	940	1204											
.IF	514														
.LIST	310	451													
.MACR	403	434	435	436	437	438	439	440	441	442	443	445	446	447	448
	449														
.MACRO	444														
.NLIST	311	451													
.PAGE	705														
.REM	19														
.REPT	451														
.TITLE	313	705													

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DZRPCB.SEG/SOL/CRF/PAGNUM=DZRPCB.M01,DZRPCB  
 RUN-TIME: 5 8 2 SECONDS  
 RUN-TIME RATIO: 152/16=9.3  
 CORE USED: 8K (15 PAGES)

