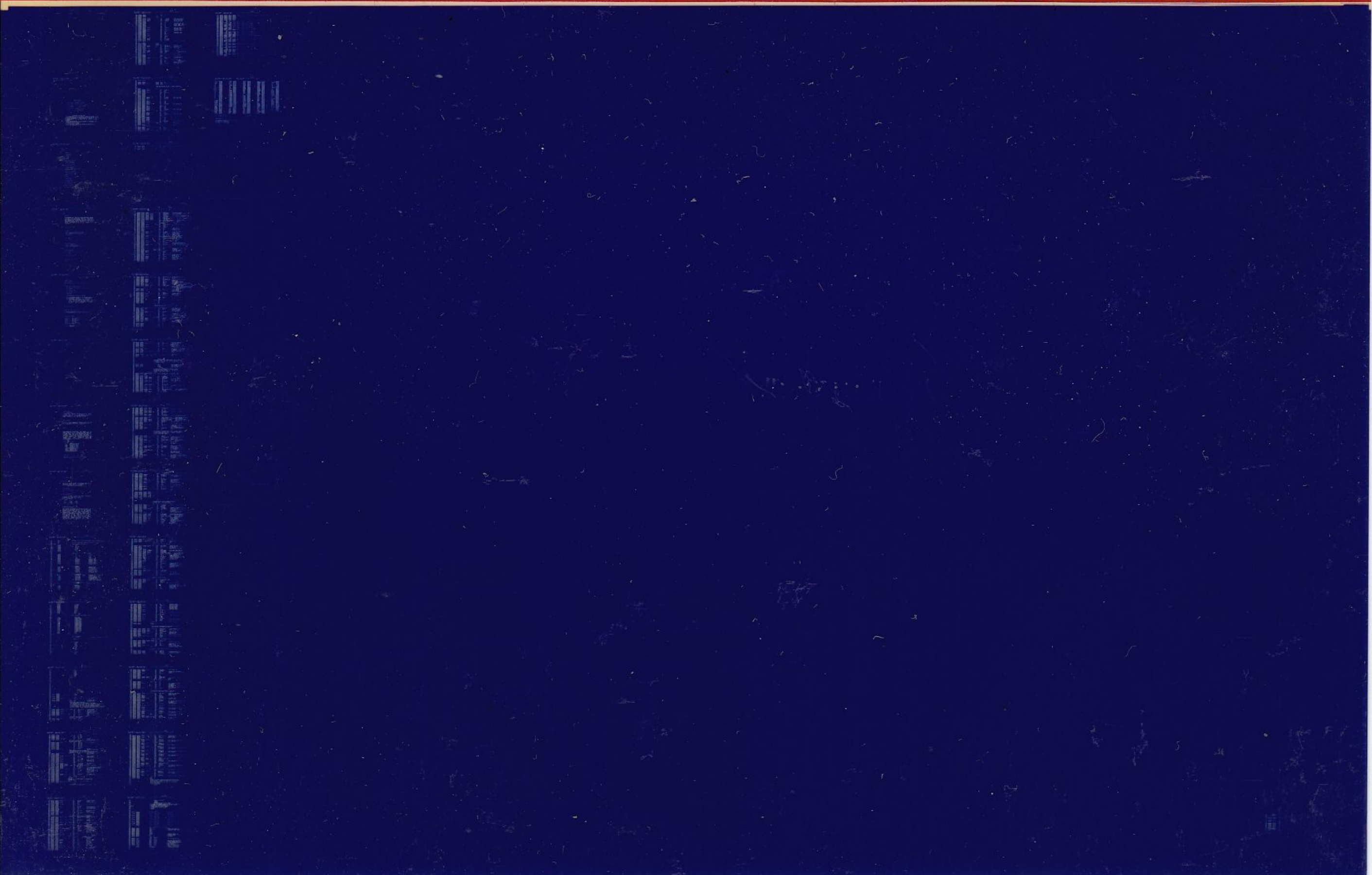


# RP11C/RP03

MULTIDRIVE DIAGNOSTIC  
MD-11-DZRPC-C

EP-DZRPC-C-DL-A  
COPYRIGHT © 72-77  
FICHE 1 OF 1

AUG 1977  
**digital**  
MADE IN USA



B01

ECF:02RPBESQ

00010000

770712

POP10 411

HDR10ZRPCCSEQ

00010000

770712

MD-11-DZRPC-A, RP11C MULTI-DRIVE  
DZRPC.P11 18-MAY-77 12:50

MACY11 30(1046) 06-JUN-77 14:48 PAGE 2

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZRPC-C-D  
PRODUCT NAME: RP11C MULTI DRIVE DIAGNOSTIC  
DATE RELEASED: APRIL, 1977  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: JOE STUBBLEBINE

COPYRIGHT (C) 1972, 1977  
THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT  
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL  
EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES  
NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS  
DOCUMENT.  
THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A  
LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH  
THE TERMS OF SUCH LICENSE.  
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  
FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT  
THAT IS NOT SUPPLIED BY DIGITAL.

CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS
  - 2.1 EQUIPMENT
  - 2.2 STORAGE
  - 2.3 PRELIMINARY PROGRAMS
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
  - 4.1 CONTROL SWITCH SETTINGS
  - 4.2 STARTING ADDRESS
  - 4.3 PROGRAM AND/OR OPERATOR ACTION
- 5.0 OPERATING PROCEDURE
  - 5.1 OPERATION SWITCH SETTINGS
  - 5.2 SUBROUTINE ABSTRACT
- 6.0 ERRORS
- 7.0 RESTRICTIONS
- 8.0 MISCELLANEOUS
  - 8.1 EXECUTION TIME
  - 8.2 STACK POINTER
  - 8.3 ERROR INFORMATION
- 9.0 PROGRAM DESCRIPTION

1. ABSTRACT

THIS PROGRAM WILL TEST UP TO EIGHT RP02/RP03 DRIVES ON AN RP11C 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  
RP11C DISK CONTROLLER WITH UP TO EIGHT RP02/RP03  
DISK DRIVES

2.2 STORAGE

4K OF STORAGE IS REQUIRED TO RUN THIS TEST

2.3 PRELIMINARY PROGRAMS

DZRPA DISKLESS DIAGNOSTIC  
DZRPB DISK RELIABILITY DIAGNOSTIC

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1

## 4.2 STARTING ADDRESS

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.  
(NOTE: THE FIRST PASS, A 'QUICK VERIFY' PASS, PERFORMS EACH OF THE FOUR TEST FUNCTIONS ON EACH DRIVE ONLY ONCE. THIS IS TO ENSURE THAT THE DRIVES ARE BASICALLY 'THERE' AND RUNNING.)

## 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

GO1

MC-11-DZRPC-A. RP11C MULTI-DRIVE  
DZRPC.P11 18-MAY-77 12:50

MACY11 30(1046) 06-JUN-77 14:48 PAGE 6

5.2 SUBROUTINE ABSTRACTS

5.2.1 HLT

THIS ROUTINE IS ENTERED UPON DETECTION OF AN ERROR. 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

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.  
(SEE NOTE IN 4.3 FOR EXPLANATION OF QV PASS)

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 SEFK 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.

%

; COPYRIGHT 1972, 1977 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

; CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

; GENERAL REGISTER ASSIGNMENTS

```

319
320
321
322
323
324      000000      R0=%0
325      000001      R1=%1
326      000002      R2=%2
327      000003      R3=%3
328      000004      R4=%4
329      000005      R5=%5
330      000006      SP=%6
331      000007      PC=%7
  
```

; STATUS REGISTER (PSW) BIT ASSIGNMENTS

```

332
333
334      000001      C=1          ; C BIT
335      000002      V=2          ; V BIT
336      000004      Z=4          ; Z BIT
337      000010      N=10         ; N BIT
338      000020      T=20         ; T BIT
339      000340      PRI7=340     ; PRIORITY LEVEL 7
340      000300      PRI6=300     ; PRIORITY LEVEL 6
341      000240      PRI5=240     ; PRIORITY LEVEL 5
342      000200      PRI4=200     ; PRIORITY LEVEL 4
343      000140      PRI3=140     ; PRIORITY LEVEL 3
344      000100      PRI2=100     ; PRIORITY LEVEL 2
345      000040      PRI1=40      ; PRIORITY LEVEL 1
  
```

; VECTOR ADDRESSES

```

346
347
348      000004      ERRVEC=4      ; ERROR VECTOR
349      000010      RESVEC=10     ; RESERVED INST VECTOR
350      000014      TBITVEC=14    ; T BIT VECTOR
351      000020      IOTVEC=20     ; IOT TRAP VECTOR
352      000024      PFVEC=24      ; POWER FAIL VECTOR
353      000030      EMTVEC=30     ; EMT VECTOR
354      000034      TRAPVEC=34    ; TRAP VECTOR
  
```

; REGISTER ADDRESSES

```

355
356
357      177776      PSW=177776    ; PROCESSOR STATUS REGISTER
358      177560      TKS=177560    ; KEYBOARD CSR
359      177562      TKB=177562    ; ADDR OF KEYBOARD BUFFER
360      177564      TPS=177564    ; TELEPRINTER CSR
361      177566      TPB=177566    ; TELEPRINTER BUFFER
362      177570      SWR=177570    ; CONSOLE SWITCH REGISTER
363      177570      DISPLAY=177570 ; CONSOLE DISPLAY REGISTER
  
```

; INITIAL STACK POINTER

```

364
365
366      000500      STKPTR=500     ; PROGRAM STACK PCINTER
  
```

; BIT ASSIGNMENTS

```

367
368
369      100000      B15=100000
370      040000      B14=40000
371      020000      B13=20000
372      010000      B12=10000
373      004000      B11=4000
  
```

374	002000
375	001000
376	000400
377	000200
378	000100
379	000040
380	000020
381	000010
382	000004
383	000002
384	000001
385	
386	
387	
388	177572
389	172340
390	172342
391	172344
392	172346
393	172350
394	172352
395	172354
396	172356
397	172300
398	172302
399	172304
400	172306
401	172310
402	172312
403	172314
404	172316
405	000006
406	000000
407	
408	
409	
410	104400
411	
412	104000
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	
466	
467	
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	

B10=2000  
 B9=1000  
 B8=400  
 B7=200  
 B6=100  
 B5=40  
 B4=20  
 B3=10  
 B2=4  
 B1=2  
 B0=1

;MEMORY MANAGEMENT REGISTER ASSIGNMENTS

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

410 104400  
 412 104000

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

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  
483  
484  
485

: HLT  
: PRINT  
: DUMP  
: DUMPF  
: SDUMP  
: SDUMPF  
: RAND  
: READ  
: PACK

000046  
000046 003002  
000052 000000  
000200 012707 002350  
001000 000000  
001002 000000  
001004 032737 040000 177570  
001012 001403  
001014 005767 000232  
001020 001003  
001022 005067 000224  
001026 000002  
001030 016716 177746  
001034 000002  
001036 012667 000020  
001042 010546  
001044 010446

. =46  
\$ENDAD  
. =52  
0  
. =200  
MOV #START,PC ;GO TO START OF TEST  
. =1000  
ICNT: 0 ;CONTAINS PASS COUNT  
LAD: 0 ;PROGRAM TRACE  
;SCOPE (EMT) SERVICE ROUTINE  
;THIS ROUTINE WILL LOOP IF AN ERROR OCCURED AND  
;LOOP ON ERROR SWITCH IS SET (BIT 14). IF LOOPING IS INDICATED  
;THE CONTENTS OF "LAD" EQUAL THE LOOP ADDRESS. IN ORDER  
;TO LOOP ON ERROR, BIT 14 OF THE SWITCH REGISTER MUST BE SET AND  
;LOCATION "ERRFLG" MUST BE NEGATIVE INDICATING AN ERROR. ONCE THE  
;LOOP IS INITIATED IT WILL CONTINUE UNTIL SWITCH 14 IS CLEARED.  
SCOPES: BIT #B14,2#SWR ;LOOP ON ERROR?  
BEQ 2\$ ;BRANCH IF NO  
TST ERRFLG ;IS THERE AN ERROR?  
BNE 1\$ ;BRANCH IF YES  
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)



```

542 001276 032737 020000 177570 PRINTS: BIT      #B13,@#SWR      ;INHIBIT TYPEOUTS?
543 001304 001403          BEQ      PRINT$      ;BRANCH IF NO
544 001306 062705 000002          ADD      #2,R5      ;UPDATE RETURN ADDR
545 001312 000205          RTS      R5
546 001314 105737 177564          PRNTP$: TSTB   @#TPS      ;WAIT FOR PRINTER TO FINISH
547 001320 100375          BPL      #-4
548 001322 010546          MOV      R5,-(SP)
549 001324 062716 000002          ADD      #2,(SP)      ;ADJUST RETURN PC
550 001330 011505          MOV      (R5),R5      ;GET MESSAGE ADDR
551 001332 105715          1$: TSTB   (R5)      ;CHECK FOR TERMINATOR
552 001334 001002          BNE      2$
553 001336 012605          MOV      (SP)+,R5      ;GET RETURN ADDR
554 001340 000205          RTS      R5      ;RETURN
555 001342 112537 177566          2$: MOV     (R5)+,@#TPB      ;PRINT CHARACTER
556 001346 105737 177564          TSTB   @#TPS      ;WAIT TILL DONE
557 001352 100375          BPL      #-4
558 001354 000766          BR      1$
559          ;THIS ROUTINE TYPES A LOCATION IN OCTAL
560 001356 032737 020000 177570 PRINTR: BIT      #B13,@#SWR      ;INHIBIT TYPEOUT?
561 001364 001406          BEQ      PRINTA      ;BRANCH IF NO
562 001366 000207          RTS      PC
563 001370 032737 020000 177570 PRINTS: BIT      #B13,@#SWR      ;INHIBIT TYPEOUT?
564 001376 001405          BEQ      PRINTB      ;BRANCH IF NO
565 001400 000207          RTS      PC
566 001402 112767 000001 000140 PRINTA: MOV     #1,.PR      ;SET ZERO FILL SWITCH
567 001410 000402          BR      .+6      ;SKIP
568 001412 005067 000132          PRINTB: CLR     .PR      ;SUPPRESS LEADING ZEROS
569 001416 112767 177772 000125 .PTIT: MOV     #-6,.PR+1      ;SET COUNT
570 001424 010446          .PTIT: MOV     R4,-(SP)      ;SAVE R4
571 001426 012704 001552          MOV     #.PR+2,R4      ;SET POINTER TO FIRST CHARACTER
572 001432 105014          CLRB   (R4)      ;CLEAR FIRST BYTE
573 001434 000413          BR      .PRF      ;ROTATE FIRST BIT
574 001436 105014          .PRL: CLRB   (R4)      ;CLEAR BYTE OF CHAR
575 001440 032767 000100 000102 BIT      #100,.PR      ;BIT TYPING MODE
576 001446 001006          BNE     .PRF      ;YES SKIP 2 ROTATES
577 001450 006167 000120          ROL     TTY      ;ROTATE BIT INTO C
578 001454 106114          ROLB   (4)      ;PACK IT
579 001456 006167 000112          ROL     TTY
580 001462 106114          ROLB   (4)
581 001464 006167 000104          .PRF: ROL     TTY
582 001470 106114          ROLB   (4)
583 001472 105714          TSTB   (4)      ;IS IT ZERO
584 001474 001402          BEQ     .+6      ;SKIP INC
585 001476 105267 000046          INCB   .PR      ;SET FILL SWITCH
586 001502 105767 000042          TSTB   .PR      ;CHECK FILL SWITCH
587 001506 001402          BEQ     .+6      ;SKIP BITSET
588 001510 152724 000060          BISB   #'0,(4)+      ;MAKE INTO ASCIZ CHAR
589 001514 105267 000031          INCB   .PR+1      ;INC COUNT
590 001520 001346          BNE     .PRL      ;REPEAT
591 001522 022704 001552          CMP     #.PR+2,R4      ;EMPTY BUFFER
592 001526 001002          BNE     .+6      ;SKIP IF NOT
593 001530 112724 000060          MOV     #'0,(4)+      ;LOAD ONE ZERO
594 001534 105014          CLRB   (4)      ;NULL TERMINATOR
595 001536 004567 177534          JSR     R5,PRINT$      ;PRINT MESSAGE
596 001542 001552          .PR+2
597 001544 012604          MOV     (SP)+,R4      ;RESTORE R4

```

598	001546	000207			RTS	PC	
599	001550	000012			.PR:	.BLKW	12
600	001574	000000			TTY:	0	
601	001576				RANDS:		
602	001576	004767	177234		JSR	PC,SAVES	;SAVE THE REGISTERS
603	001602	016700	000106		MOV	LONUM,R0	;SET R0 WITH LOW
604	001606	016701	000100		MOV	HINUM,R1	;SET R1 WITH HIGH
605	001612	012703	177771		MOV	#-7,R3	;SET SHIFT COUNT
606	001616	005002			CLR	R2	
607	001620	006300		1S:	ASL	R0	;SHIFT R0 LEFT AND
608	001622	006101			ROL	R1	;ROTATE CARRY INTO R1 AND
609	001624	006102			ROL	R2	;ROTATE CARRY INTO R2
610	001626	005203			INC	R3	;CHECK FOR DONE
611	001630	001373			BNE	1S	
612	001632	066702	000056		ADD	LONUM,R2	;ADD # TO MAKE X 129
613	001636	005501			ADC	R1	;PROPOGATE CARRY
614	001640	066701	000046		ADD	HINUM,R1	;ADD # TO MAKE X 129
615	001644	005502			ADC	R2	;PROPOGATE CARRY
616	001646	062700	001057		ADD	#1057,R0	
617	001652	005501			ADC	R1	;PROPOGATE CARRY
618	001654	005502			ADC	R2	;PROPOGATE CARRY
619	001656	062701	047401		ADD	#47401,R1	
620	001662	005502			ADC	R2	
621	001664	062702	000006		ADD	#6,R2	
622	001670	060200			ADD	R2,R0	
623	001672	005501			ADC	R1	
624	001674	010067	000014		MOV	R0,LONUM	
625	001700	010167	000006		MOV	R1,HINUM	
626	001704	004767	177154		JSR	PC,RESTS	;RESTORE THE REGISTERS
627	001710	000207			RTS	PC	
628							
629	001712	000000			HINUM:	0	
630	001714	000000			LONUM:	0	
631	001716	010346			READS:	MOV	R3,-(6)
632	001720	012703	002026		1S:	MOV	#INPUTS,R3
633	001724	022703	002046		2S:	CMP	#INPUTS+20,R3
634	001730	001412				BEG	4S
635	001732	105737	177560		TSTB	#177560	;WAIT FOR A CHAR
636	001736	100375			BPL	-4	
637	001740	113713	177562		MOV	#177562,(3)	;GET CHAR
638	001744	142713	000200		BICB	#200,(3)	;GET RID OF JUNK
639	001750	122713	000177		CMPB	#177,(3)	;IS IT A RUBOUT?
640	001754	001004			BNE	3S	;SKIP IF NO
641	001756			4S:			
642	001756	004567	177314		JSR	RS,PRINTS	;PRINT MESSAGE
643	001762	002066			READMS		
644	001764	000755			BR	1S	;CLEAR BUFFER AND START OVER
645	001766	013737	177562	177566	3S:	MOV	#TKB,#*PB
646	001774	105737	177564		TSTB	#TPS	;ECHO THE CHAR
647	002000	100375			BPL	-4	;WAIT FOR READY
648	002002	122723	000015		CMPB	#15,(3)+	;CHECK FOR RETURN
649	002006	001346			BNE	2S	;LOOP IF NOT RETURN
650	002010	105063	177777		CLRB	-1(3)	;REMOVE THE RETURN
651	002014	004567	177256		JSR	RS,PRINTS	;PRINT MESSAGE
652	002020	002072			READLS		
653	002022	012603			MOV	(6)+,R3	;RESTORE R3

```

654 002024 000207          RTS      PC          ;RETURN
655
656 002026 000020          INPUTS: .BLKW 20
657 002066 006477 000012      READMS: .ASCIZ '?'(15)(12)
658 002072 000012      READLS: .ASCIZ (12)
659
660          ;TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
661          ;PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
662
663 002074          PACKS:
664 002074 004767 176736      JSR      PC,SAVES      ;SAVE THE REGISTERS
665 002100 005067 000242      CLR      NUMS
666 002104 005000          CLR      RO
667 002106 105760 002026      2S:    TSTB     INPUTS(RO)
668 002112 001402          BEQ      1S
669 002114 005200          INC      RO
670 002116 000773          BR       2S
671 002120 005300          1S:    DEC      RO
672 002122 004767 000166      JSR      PC,PACS      ;GET OCTAL CHAR
673 002126 016767 000212 000212      MOV      PK$,NUMS     ;PACK FIRST CHAR
674 002134 004767 000154          JSR      PC,PACS      ;GET OCTAL CHAR
675 002140 000241          CLC
676 002142 006167 000176      ROL      PK$
677 002146 006167 000172      ROL      PK$
678 002152 006167 000166      ROL      PK$
679 002156 056767 000162 000162      BIS      PK$,NUMS     ;PACK SECOND CHAR
680 002164 004767 000124          JSR      PC,PACS      ;GET OCTAL CHAR
681 002170 000241          CLC
682 002172 000367 000146      SWAB     PK$
683 002176 006067 000142      ROR      PK$
684 002202 006067 000136      ROR      PK$
685 002206 056767 000132 000132      BIS      PK$,NUMS     ;PACK THIRD CHAR
686 002214 004767 000074          JSR      PC,PACS      ;GET OCTAL CHAR
687 002220 000367 000120      SWAB     PK$
688 002224 000241          CLC
689 002226 006167 000112      ROL      PK$
690 002232 056767 000106 000106      BIS      PK$,NUMS     ;PACK FOURTH CHAR
691 002240 004767 000050          JSR      PC,PACS      ;GET OCTAL CHAR
692 002244 000367 000074      SWAB     PK$
693 002250 000241          CLC
694 002252 006167 000066      ROL      PK$
695 002256 006167 000062      ROL      PK$
696 002262 006167 000056      ROL      PK$
697 002266 006167 000052      ROL      PK$
698 002272 056767 000046 000046      BIS      PK$,NUMS     ;PACK FIFTH CHAR
699 002300 000402          BR
700 002302 062706 000002      PKEXS:  ADD      #2,SP      ;MODIFY STACK
701 002306          PKEX1$:
702 002306 004767 176552      JSR      PC,RESTS     ;RESTORE THE REGISTERS
703 002312 000207          RTS      PC          ;EXIT
704
705 002314 005700          PACS:  TST      RO
706 002316 100771          BMI     PKEXS
707 002320 005067 000020          CLR      PK$
708 002324 116067 002026 000012      MOVB     INPUTS(RO),PK$ ;GET INPUT CHAR
709 002332 005300          DEC      RO

```



MD-11-DZRPC-A, RP11C MULTI-DRIVE  
DZRPC.P11 18-MAY-77 12:50

MACY11 30(1046) 06-JUN-77 14:48 PAGE 16

710 002334 042767 177770 000002  
711 002342 000207  
712  
713 002344 000000  
714 002346 000000  
715

BIC #177770,PKS :CLEAR UNWANTED BITS  
RTS PC  
PKS: 0  
NUMS: 0

```

716
717 002350 012706 000500 START: MOV #STKPTR,SP ;SET STACK POINTER
718 002354 012737 000340 177776 MOV #340,#PSW ;LOCK UP INTERRUPTS
719 002362 012767 001112 175444 MOV #ERROR,34 ;SETUP ERROR TRAP
720 002370 012767 000340 175440 MOV #PRI7,36
721 002376 012767 001004 175424 MOV #SCOPE$,30 ;SETUP SCOPE TRAP
722 002404 012767 000340 175420 MOV #PRI7,32
723 002412 012737 005016 000254 MOV #DSKINT,#VECTOR ;SET UP DISK INTERRUPT VECTOR
724 002420 012737 000340 000256 MOV #340,#STATUS
725 002426 023737 000042 000046 CMP #42,#46 ;ARE WE IN ACT11 AUTO MODE?
726 002434 001403 BEQ 1$ ;YES, SKIP TITLE
727 002436 004567 176634 JSR R5,PRINT$ ;PRINT MESSAGE
728 002442 005726 TITLE
729 002444 005000 1$: CLR R0
730 002446 005060 005460 CLRTAB: CLR DEVTBL(R0) ;CLEAR THE DEVICE TABLE
731 002452 005720 TST (R0)+
732 002454 022700 000200 CMP #128,R0
733 002460 001372 BNE CLRTAB
734 002462 005737 000042 TST #42 ;UNDER MONITOR CONTROL?
735 002466 001424 BEQ 5$ ;BRANCH IF NO
736 002470 005000 CLR R0 ;CLEAR MODIFIER
737 002472 005001 CLR R1
738 002474 012777 000001 003202 7$: MOV #1,#RPCS ;CLEAR THE CONTROLLER
739 002502 110177 003200 MOV# R1,#RPCS1 ;SELECT UNIT
740 002506 005777 003210 TST #RPCS ;IS UNIT READY?
741 002512 100403 BMI 6$ ;BRANCH IF YES
742 002514 052760 100000 005460 BIS #B15,DEVTBL(R0) ;SET UNIT UNAVAILABLE BIT
743 002522 062700 000020 6$: ADD #16,R0 ;UPDATE MODIFIER
744 002526 005201 INC R1 ;UPDATE UNIT NUMBER
745 002530 032701 000010 BIT #B3,R1 ;ALL UNITS TESTED?
746 002534 001757 BEQ 7$ ;BRANCH IF NO
747 002536 000420 BR 8$
748 002540 012701 000001 5$: MOV #1,R1
749 002544 005000 CLR R0
750 002546 030137 177570 2$: BIT R1,#SWR ;TEST THE SWITCH REGISTER
751 002552 001003 BNE 1$ ;TO DETERMINE WHICH UNITS
752 002554 052760 100000 005460 BIS #B15,DEVTBL(R0) ;TO TEST. IF THE UNIT IS UNAVAILABLE
753 002562 062700 000020 1$: ADD #16,R0 ;SET BIT 15 IN THE DEVICE TABLE
754 002566 000241 CLC
755 002570 006101 ROL R1
756 002572 032701 000400 BIT #B8,R1 ;HAVE ALL UNITS BEEN SCANNED?
757 002576 001763 BEQ 2$ ;NO-BRANCH
758 002600 000005 8$: RESET ;CLEAR THE SYSTEM
759 002602 004567 001746 JSR R5,EXTMEN ;DETERMINE AMOUNT OF CORE
760 002606 005067 003046 CLR UNIT ;INITIALIZE POINTER
761 002612 005067 003044 CLR PASSCT
762 002616 005005 CLR R5
763 002620 032765 100000 005460 4$: BIT #B15,DEVTBL(R5) ;IS UNIT AVAILABLE?
764 002626 001002 BNE 3$ ;BRANCH IF NO
765 002630 004767 000160 JSR PC,HOME ;DO A HOME SEEK
766 002634 005267 003020 3$: INC UNIT ;UPDATE UNIT
767 002640 062705 000020 ADD #16,R5 ;UPDATE TABLE POINTER
768 002644 032767 000010 003006 BIT #B3,UNIT ;HAVE ALL UNITS BEEN HOMED?
769 002652 001762 BEQ 4$ ;NO-BRANCH
770 002654 005067 003000 LOOP: CLR UNIT
771 002660 005005 CLR R5

```

```

772 002662 032765 100000 005460 MAIN: BIT #B15,DEVTBL(R5) ;IS THE UNIT AVAILABLE?
773 002670 001004 BNE 1$ ;BRANCH IF NO
774 002672 016504 005460 MOV DEVTBL(R5),R4
775 002676 004774 003172 JSR PC,JMPTBL(R4) ;PERFORM FUNCTION IN JMPTBL
776 002702 005267 002752 1$: INC UNIT ;UPDATE UNIT
777 002706 062705 000020 ADD #16.,R5 ;UPDATE TABLE POINTER
778 002712 032767 000010 002740 BIT #B3,UNIT ;HAVE ALL UNITS BEEN SCANNED?
779 002720 001760 BEQ MAIN ;NO BRANCH
780 002722 005267 002734 INC PASSCT ;INCREMENT ITERATION COUNTER
781 002726 016737 002730 177570 MOV PASSCT,#SWR ;DISPLAY COUNT
782 002734 022767 000004 002720 CMP #4,PASSCT ;TEST TO SEE IF THIS IS THE FIRST TIME
783 ;WE HAVE DONE THE FOUR TEST FUNCTIONS
784 ;ON EACH DRIVE (SEEK,SEKCK,WRITE,READ).
785 002742 001004 BNE 2$ ;BRANCH IF NO
786 002744 004567 176326 JSR R5,PRINT$ ;PRINT MESSAGE
787 002750 006205 MES20 ;GIVES A QUICK INDICATION THAT THE UNITS
788 ;ARE UP & RUNNING.
789 002752 000404 BR 2$+10
790 002754 022767 005000 002700 2$: CMP #5000,PASSCT ;IS PASS COMPLETE?
791 002762 001013 BNE MEXIT1 ;BRANCH IF NO
792 002764 004567 176306 JSR R5,PRINT$ ;PRINT MESSAGE
793 002770 006235 MES21
794 002772 013701 000042 MOV #42,R1 ;GET RETURN ADDRESS
795 002776 001405 BEQ MEXIT1
796 003000 000005 RESET
797 003002 004711 SENDAD: JSR PC,(R1) ;RETURN TO MONITOR
798 003004 000240 NOP
799 003006 000240 NOP
800 003010 000240 NOP
801 003012 000720 MEXIT1: BR LOOP ;LOOP
802
803 ;THIS ROUTINE WILL SEEK HOME THE PACK WHOSE
804 ;ADDRESS IS IN UNIT.
805
806 003014 116777 002640 002664 HOME: MOVB UNIT,DRPCS1 ;LOAD THE UNIT #
807 003022 005777 002674 TST DRPDS ;IS THE UNIT READY?
808 003026 100401 BMI 5$ ;BRANCH IF READY
809 003030 104400 HLT ;UNIT IS NOT READY
810 003032 112777 000015 002644 5$: MOVB #15,DRPCS ;DO A HOME SEEK
811 003040 012794 000025 MOV #25,R4
812 003044 005304 1$: DEC R4 ;WAIT FOR SEEK TO START
813 003046 001376 BNE 1$
814 003050 032777 002000 002644 BIT #B10,DRPDS ;IS SEEK UNDER WAY SET
815 003056 001001 BNE 2$ ;YES
816 003060 104400 HLT ;SEEK UNDERWAY DID NOT SET
817 003062 016704 002572 2$: MOV UNIT,R4
818 003066 005067 000066 CLR ATTN ;
819 003072 116467 003162 000060 MOVB ATTN(R4),ATTNB ;DETERMINE ATTENTION RESPONSE
820 003100 012701 000005 MOV #5,R1 ;BEGINNING OF DELAY
821 003104 005301 DEC R1 ;LOOP TO ALLOW DRIVE
822 ;TIME TO BECOME READY
823 003106 005000 CLR R0
824 003110 005200 7$: INC R0
825 003112 001376 BNE 7$
826 003114 005701 6$: TST R1
827 003116 001372 BNE .-12

```

```

828 003120 005200          INC      RD          ;TIME OUT ATTENTION BIT
829 003122 036777 000032 002572  BIT     ATTNB,DRPDS ;DID ATTENTION SET?
830 003130 001003          BNE     3$          ;BRANCH IF YES
831 003132 005700          TST     RD          ;DID IT TIME OUT?
832 003134 001367          BNE     6$          ;BRANCH IF NO
833 003136 104400          HLT                    ;ATTENTION BIT DID NOT SET
834
835 003140 005777 002540 3$:  TST     DRPCS        ;ANY DEVICE STATUS ERRORS?
836 003144 100001          BPL     4$          ;NO-BRANCH
837 003146 104400          HLT                    ;DEVICE STATUS ERROR AFTER HOME COMMAND
838 003150 116777 000004 002544 4$:  MOVB    ATTNB,DRPDS ;CLEAR ATTENTION BIT
839 003156 000207          RTS     PC          ;EXIT
840
841 003160 000000          ATTNB: 0            ;CONTAINS ATTENTION BIT FOR CURRENT UNIT
842 003162 001 002 004  ATTN:  .BYTE 1,2,4,10,20,40,100,200
843 003165 010 020 040
844 003170 100 200
845
846          .EVEN
847          ;THIS TABLE DETERMINES WHERE CONTROL WILL GO FOR ANY
848          ;PARTICULAR UNIT. THE FIRST WORD OF THE DEVICE TABLE
849          ;IS USED AS A MODIFIER FOR A JSR INDIRECT INTO
850          ;THE FOLLOWING TABLE.
851
852 003172 003202          JMPTBL: SEEK        ;SEEK A RANDOM CYLINDER
853 003174 003516          SEKCK              ;SEE IF SEEK IS COMPLETE
854 003176 004030          WRITE            ;WRITE RANDOM DATA
855 003200 004204          READD           ;READ AND VERIFY RANDOM DATA
856
857          ;THIS ROUTINE WILL GENERATE A RANDOM CYLINDER
858          ;ADDRESS AND ISSUE A SEEK TO IT. THE FUNCTION
859          ;POINTER IN THE DEVICE TABLE WILL BE UPDATED TO
860          ;CHECK FOR THE ATTENTION BIT.
861
862 003202 016565 005464 005462  SEEK:  MOV     DEVTBL+4(R5),DEVTBL+2(R5) ;SET UP CYLINDER FROM ADDRESS
863 003210 116777 002444 002470  MOVB    UNIT,DRPCS1 ;SELECT THE UNIT
864 003216
865 003216 004767 176354          1$:  JSR     PC,RAND$    ;GENERATE TWO RANDOM NOS.
866 003222 016767 176466 002436  MOV     LOUM,WORK1
867 003230 016767 176456 002432  MOV     HINUM,WORK2
868 003236 032777 020000 002456  BIT     #B13,DRPDS ;IS THIS AN RPO3?
869 003244 001410          BEQ     6$          ;BRANCH IF NO
870 003246 042767 177000 002412  BIC     #177000,WORK1
871 003254 022767 000625 002404  CMP     #625,WORK1 ;GENERATE A CYLINDER ADDRESS
872 003262 002755          BLT     1$
873 003264 000407          BR      7$
874 003266 042767 177400 002372 6$:  BIC     #177400,WORK1
875 003274 022767 000312 002364  CMP     #312,WORK1 ;IS NUMBER TOO LARGE?
876 003302 002745          BLT     1$          ;BRANCH IF YES
877 003304 026765 002356 005462 7$:  CMP     WORK1,DEVTBL+2(R5)
878 003312 001741          BEQ     1$
879 003314 016765 002346 005464  MOV     WORK1,DEVTBL+4(R5) ;SAVE CYLINDER ADDRESS
880 003322 016765 002342 005466  MOV     WORK2,DEVTBL+6(R5) ;USE A RANDOM DATA BASE
881 003330 004767 176242          JSR     PC,RAND$    ;GENERATE TWO RANDOM NOS.
882 003334 016767 176354 002324  MOV     LOUM,WORK1
883 003342 016767 176344 002320  MOV     HINUM,WORK2

```

```

884 003350 042767 177760 002310      BIC      #177760,WORK1
885 003356 022767 000011 002302      CMP      #11,WORK1      ;GENERATE A RANDOM SECTOR
886 003364 002003                BGE      2$
887 003366 162767 000010 002272      SUB      #8,WORK1
888 003374 042767 177740 002266      BIC      #177740,WORK2 ;GENERATE A RANDOM TRACK
889 003402 022767 000023 002260      CMP      #23,WORK2
890 003410 002003                BGE      3$
891 003412 162767 000014 002250      SUB      #14,WORK2
892 003420 116767 002244 002241      MOV      WORK2,WORK1+1 ;MERGE TRACK AND SECTOR ADDR
893
894 003426 016765 002234 005470      MOV      WORK1,DEVTBL+10(R5) ;STORE RANDOM DISK ADDRESS
895 003434 005065 005472                CLR      DEVTBL+12(R5) ;CLEAR TIMEOUT COUNTER
896 003440 016577 005464 002246      MOV      DEVTBL+4(R5),#ARPCA ;LOAD CYLINDER ADDRESS
897 003446 016577 005470 002242      MOV      DEVTBL+10(R5),#ARPCDA ;LOAD TRACK AND SECTOR
898 003454 112777 000011 002222      MOV      #11,#ARPCS ;INITIATE A SEEK
899 003462 012704 000025                MOV      #25,R4
900 003466 005304                4$: DEC      R4 ;WAIT FOR SEEK TO START
901 003470 001376                BNE      4$
902 003472 032777 002000 002222      BIT      #810,#ARPCDS ;IS THE SEEK UNDERWAY?
903 003500 001001                BNE      5$ ;YES-BRANCH
904 003502 104400                HLT
905 003504 005265 005460      5$: INC      DEVTBL(R5) ;SEEK UNDERWAY DID NOT SET
906 003510 005265 005460      INC      DEVTBL(R5) ;MODIFY FUNCTION POINTER TO
907 003514 000207                RTS      PC ;CHECK FOR SEEK COMPLETE
908 ;EXIT
909 ;THIS ROUTINE IS ENTERED AFTER A SEEK HAS BEEN ISSUED.
910 ;IT CHECKS THE ATTENTION FLAG - IF CLEAR IT UPDATES THE
911 ;TIMEOUT COUNTER AND CHECKS IT. IF SET IT VERIFIES
912 ;THE SEEK FUNCTIONED PROPERLY.
913
914 003516 016704 002136      SEKCK: MOV      UNIT,R4
915 003522 116467 003162 177430      MOV      ATTN(R4),ATTNB ;DETERMINE ATTENTION BIT
916 003530 036777 177424 002164      BIT      ATTNB,#ARPCDS ;TEST FOR ATTENTION FLAG
917 003536 001014                BNE      1$ ;BRANCH IF SET
918 003540 005265 005472                INC      DEVTBL+12(R5) ;UPDATE TIMEOUT COUNTER
919 003544 022765 005000 005472      CMP      #5000,DEVTBL+12(R5) ;DID OPERATION TIMEOUT?
920 003552 101005                BHI      2$ ;NOT YET-BRANCH
921 003554 116777 002100 002124      MOV      UNIT,#ARPCS1 ;SELECT UNIT
922 003562 104400                HLT ;SEEK TIMED OUT
923 003564 000447                BR      4$
924 003566 000207                2$: RTS      PC ;EXIT
925 003570 116777 177364 002124      1$: MOV      ATTNB,#ARPCDS ;CLEAR ATTENTION FLAG
926 003576 116777 002056 002102      MOV      UNIT,#ARPCS1 ;SELECT UNIT
927 003604 032777 002000 002110      BIT      #810,#ARPCDS ;IS SEEK UNDERWAY CLEAR?
928 003612 001402                BEQ      3$ ;IF YES-BRANCH
929 003614 104400                HLT ;SEEK UNDERWAY DID NOT CLEAR
930 003616 000432                BR      4$
931 003620 005777 002060      3$: TST      #ARPCS ;ARE THERE ANY DEVICE STATUS ERRORS?
932 003624 100002                BPL      5$ ;BRANCH-NO ERRORS
933 003626 104400                HLT ;DEVICE STATUS ERRORS
934 003630 000425                BR      4$
935 003632 017704 002066      5$: MOV      #ASUCA,R4 ;GET CURRENT CYLINDER ADDRESS
936 003636 020465 005464      CMP      R4,DEVTBL+4(R5) ;DOES IT MATCH CYLINDER REQUESTED?
937 003642 001440                BEQ      6$ ;YES-BRANCH
938 003644 104400                HLT ;SUCA AND CYL REQUESTED DID NOT COMPARE
939 003646 004567 175424      JSR      R5,PRINT$ ;PRINT MESSAGE

```

```

940 003652 003756          MES10
941 003654 016567 005464 175712  MOV      DEVTBL+4(R5),TTY
942 003662 004767 175502  JSR      PC,PRINTS      ;TYPE LOCATION-SUPRESS ZEROS
943 003666 004567 175404  JSR      RS,PRINTS      ;PRINT MESSAGE
944 003672 004005          MES11
945 003674 010467 175674  MOV      R4,TTY
946 003700 004767 175452  JSR      PC,PRINTR      ;TYPE LOCATION WITH LEADING ZEROS
947 003704 032777 004000 002010 4$:  BIT      #B11,ARPCD     ;SEEK INCOMPLETE?
948 003712 001411          BEQ      10$,           ;BRANCH IF NO
949 003714 112777 000015 001762  MOVVB   #15,ARPCS      ;ISSUE HOME COMMAND
950 003722 105777 001756          TSTB    ARPCS          ;WAIT FOR DONE
951 003726 100375          BPL     -4
952 003730 005777 001766          TST     ARPCD          ;WAIT FOR UNIT READY
953 003734 100375          BPL     -4
954 003736 005065 005460 10$:  CLR     DEVTBL(R5)     ;CLEAR FUNCTION POINTER
955 003742 000207          RTS     PC            ;EXIT
956 003744 005265 005460 6$:   INC     DEVTBL(R5)
957 003750 005265 005460  INC     DEVTBL(R5)     ;UPDATE FUNCTION POINTER
958 003754 000207          RTS     PC            ;EXIT
959
960 003756 005015 042522 052521 MES10: .ASCIZ  <15><12>'REQUESTED CYLINDER= '
961 003764 051505 042524 020104
962 003772 054503 044514 042116
963 004000 051105 020075 000
964 004005 015 051412 041525 MES11: .ASCIZ  <15><12>'SUCA REGISTER= '
965 004012 020101 042522 044507
966 004020 052123 051105 020075
967 004026 000
968 004030          .EVEN
969
970          ;THIS ROUTINE WILL WRITE A RANDOM SECTOR ON
971          ;A RANDOM TRACK. THE CYLINDER HAS ALREADY
972          ;BEEN SELEYED BY THE SEEK ROUTINE.
973
974 004030 004767 000636  WRITE: JSR      PC,RANADR      ;GENERATE A RANDOM BUFFER ADDR
975 004034 012767 000400 001622  MOV      #400,WORK
976 004042 016701 001630  MOV      BUFF,R1
977 004046 004767 000336  JSR      PC,RANDAT      ;GENERATE A RANDOM PATTERN
978 004052 116777 001602 001626  MOVVB   UNIT,ARPCD1     ;SELECT THE UNIT
979 004060 032777 100000 001634  BIT      #B15,ARPCD     ;IS THE SELECTED UNIT READY
980 004066 001003          BNE     1$,           ;YES-BRANCH
981 004070 104400          HLT
982 004072 000167 000100          JMP     WRTER          ;SELECTED UNIT NOT READY
983          ;START SEQUENCE OVER
984 004076 012777 177400 001604 1$:  MOV      #-400,ARPCW     ;SETUP WORD COUNT REGISTER
985 004104 016777 001566 001600  MOV      BUFF,ARPCB     ;SETUP BUS ADDR REGISTER
986 004112 016577 005464 001574  MOV      DEVTBL+4(R5),ARPCA ;SET CYLINDER ADDR
987 004120 016577 005470 001570  MOV      DEVTBL+10(R5),ARPCD ;SETUP DISK ADDR.
988 004126 005067 001546          CLR     INT            ;CLEAR INTERRUPT FLAG
989 004132 012737 000200 177776  MOV      #PRI4,ARPCW     ;ALLOW INTERRUPT
990 004140 005067 001530          CLR     INTERR         ;CLEAR ERROR FLAG
991 004144 112777 000113 001532  MOVVB   #113,ARPCS      ;INITIATE WRITE WITH INTERRUPT
992 004152 004767 000612          JSR      PC,WAIT        ;TIMEOUT THE OPERATION
993 004156 005767 001512          TST     INTERR         ;ANY ERRORS?
994 004162 001005          BNE     WRTER          ;BRANCH IF YES
995 004164 005265 005460  INC     DEVTBL(R5)     ;UPDATE FUNCTION POINTER TO READ

```

```

996 004170 005265 005460          INC    DEVTBL(R5)
997 004174 000403                    BR     READD
998 004176 005065 005460          WRTER: CLR    DEVTBL(R5)      ;RESTORE FUNCTION POINTER
999 004202 000207                    RTS    PC                    ;EXIT
1000
1001          ;READ AND VERIFY THE DATA WRITTEN
1002
1003 004204 116777 001450 001474  READD:  MOVB   UNIT,DRPCS1      ;SELECT THE UNIT
1004 004212 032777 100000 001502          BIT    #B15,DRPDS      ;IS THE SELECTED UNIT READY?
1005 004220 001003                    BNE    1$              ;YES-BRANCH
1006 004222 104400                    HLT                    ;SELECTED UNIT NOT READY
1007 004224 000167 000152          JMP    RDCNT
1008 004230 012777 177400 001452  1$:   MOV    #-400,DRPWC     ;LOAD WORD COUNT REGISTER
1009 004236 016777 001434 001446          MOV    BUFF,DRPBA     ;LOAD BUS ADDR REGISTER
1010 004244 062777 001000 001440          ADD    #1000,DRPBA
1011 004252 016577 005464 001434          MOV    DEVTBL+4(R5),DRPCA ;SET CYLINDER ADDR
1012 004260 016577 005470 001430          MOV    DEVTBL+10(R5),DRPDA ;SETUP DISK ADDR.
1013 004266 005067 001406          CLR    INT             ;CLEAR INTERRUPT FLAG
1014 004272 005067 001376          CLR    INTERR          ;CLEAR ERROR FLAG
1015 004276 112777 000117 001400          MOVB   #117,DRPCS     ;INITIATE READ WITH INTERRUPT
1016 004304 004767 000460          JSR    PC,WAIT         ;TIMEOUT THE OPERATION
1017 004310 032777 040000 001366          BIT    #B14,DRPCS     ;ANY HARD ERRORS?
1018 004316 001031                    BNE    RDCNT           ;BRANCH IF YES
1019 004320 016701 001352          MOV    BUFF,R1
1020 004324 016702 001346          MOV    BUFF,R2
1021 004330 005003                    CLR    R3
1022 004332 062701 001000          ADD    #1000,R1        ;START OF PATTERN BUFFER
1023 004336 022122                    CMP    (R1)+,(R2)+    ;COMPARE DATA
1024 004340 001006                    BNE    2$              ;BRANCH-DATA DID NOT COMPARE
1025 004342 005203                    INC    R3               ;INCREMENT COUNTER
1026 004344 022703 000400          CMP    #400,R3        ;HAS BUFFER BEEN SCANNED
1027 004350 001372                    BNE    3$              ;BRANCH-NO
1028 004352 000167 000024          JMP    RDCNT
1029
1030 004356 016267 177776 001070  2$:   MOV    -2(R2),EXPS
1031 004364 016167 177776 001064          MOV    -2(R1),RECS
1032 004372 004567 174700          JSR    R5,PRINT$      ;PRINT MESSAGE
1033 004376 006100                    MES12
1034 004400 104401                    HLT    +1              ;DATA DID NOT VERIFY
1035 004402 005065 005460          RDCNT: CLR    DEVTBL(R5) ;INITIATE FUNCTION POINTER
1036 004406 000207                    RTS    PC              ;EXIT
1037
1038          ;GENERATE A RANDOM PATTERN
1039
1040 004410 016567 005466 000132  RANDAT: MOV    DEVTBL+6(R5),RAND1 ;GET RANDOM BASE
1041 004416 016567 005470 000126          MOV    DEVTBL+10(R5),RAND2
1042 004424 016700 000120          MOV    RAND1,R0
1043 004430 016704 000116          MOV    RAND2,R4
1044 004434 012703 000007          RANDA1: MOV    #7,R3   ;SETUP SHIFT COUNT
1045 004440 005002                    CLR    R2
1046 004442 006300          SHIFT: ASL    R0        ;SHIFT R0 LEFT AND
1047 004444 006104                    ROL    R4              ;ROTATE CARRY INTO LSB OF R0 INTO R4
1048 004446 006102                    ROL    R2              ;ROTATE CARRY OUT OF R4 INTO R2
1049 004450 005303                    DEC    R3              ;DECREMENT R3
1050 004452 001373                    BNE    SHIFT          ;CONTINUE LOOP
1051 004454 066700 000070          ADD    RAND1,R0       ;ADD IN # TO MAKE X129

```

```

1052 004460 005504          ADC      R4          ;PROPOGATE CARRY
1053 004462 066704 000064  ADD      RAND2,R4    ;ADD IN # TO MAKE X129
1054 004466 005502          ADC      R2          ;PROPOGATE CARRY
1055 004470 062700 001057  ADD      #1057,R0    ;ADD LOW CONSTANT
1056 004474 005504          ADC      R4          ;PROPOGATE CARRY
1057 004476 005502          ADC      R2          ;PROPOGATE AGAIN
1058 004500 062704 047401  ADD      #47401,R4   ;ADD HIGH CONSTANT
1059 004504 005502          ADC      R2
1060 004506 062702 000006  ADD      #6,R2
1061 004512 060200          ADD      R2,R0      ;REPRIME R0 WITH HIGH DIGIT
1062 004514 005504          ADC      R4
1063 004516 010067 000026  MOV      R0,RAND1
1064 004522 010021          MOV      R0,(R1)+    ;STORE DATA IN BUFFER
1065 004524 005367 001134  DEC      WORK
1066 004530 001406          BEQ      EXGEN
1067 004532 010467 000014  MOV      R4,RAND2
1068 004536 010421          MOV      R4,(R1)+    ;STORE DATA IN BUFFER
1069 004540 005367 001120  DEC      WORK
1070 004544 001333          BNE      RANDA1      ;FILL ENTIRE BUFFER
1071 004546 000207          EXGEN: RTS          PC          ;EXIT
1072
1073 004550 000000          RAND1: 0
1074 004552 000000          RAND2: 0
1075
1076          ;THIS ROUTINE DETERMINES THE TOTAL AMOUNT OF AVAILABLE
1077          ;CORE WITHOUT USING MEMORY MANAGEMENT.
1078
1079 004554 012737 000340 177776  EXTMEN: MOV      #PRI7,@#PSW    ;LOCKUP PRIORITY LEVEL
1080 004562 012767 004632 173214  MOV      #MAXREF,4          ;SETUP IO BUS TRAP
1081 004570 012767 000340 173210  MOV      #PRI7,6
1082 004576 012767 017446 000064  MOV      #17446,SAVE        ;START WITH 4K
1083 004604 005777 000060          EXREF: TST      @SAVE          ;REFERENCE MEMORY
1084 004610 022767 157446 000052  CMP      #157446,SAVE       ;TEST FOR 28K
1085 004616 001001          BNE      1$              ;BRANCH IF LESS THAN 28K
1086 004620 000407          BR      MAXRF1
1087 004622 062767 020000 000040  1$: ADD      #20000,SAVE      ;SETUP FOR NEXT REFERENCE
1088 004630 000765          BR      EXREF
1089
1090          ;ENTER HERE WHEN IO BUS ERROR OCCURS
1091
1092 004632 162767 020000 000030  MAXREF: SUB      #20000,SAVE
1093 004640 012767 000006 173136  MAXRF1: MOV      #6,4          ;RESTORE IO BUS TRAY
1094 004646 005067 173134          CLR      6
1095 004652 005737 000042          TST      @#42              ;UNDER MONITOR CONTROL?
1096 004656 001403          BEQ      1$              ;BRANCH IF NO
1097 004660 162767 005670 000002  SUB      #3000.,SAVE        ;ADJUST TOP OF CORE
1098 004666 000205          1$: RTS          R5          ;EXIT-SAVE=MAXIMUM MEMORY
1099 004670 000000          SAVE: 0                  ;HIGHEST AVAILABLE LOCATION
1100
1101          ;GENERATE A RANDOM BUFFER ADDRESS
1102
1103 004672 016704 177772          RANADR: MOV      SAVE,R4
1104 004676 162704 006250          SUB      #ENDP,R4          ;DETERMINE BUFFER SIZE
1105 004702 162704 002000          SUB      #2000,R4          ;ALLOW ROOM FOR DATA
1106 004706 004767 174664          JSR      PC,RAND$          ;GENERATE TWO RANDOM NOS.
1107 004712 016767 174776 000746          MOV      LO$UM,WORK1

```



```

1108 004720 042767 000001 000740      BIC      #80,WORK1      ;MAKE NUMBER EVEN
1109 004726 012703 100000                MOV      #100000,R3
1110 004732 020467 000730      2$:    CMP      R4,WORK1      ;ENSURE THAT THE RANDOM
1111 004736 101005                BHI      1$              ;ADDRESS FITS WITHIN AVAILABLE
1112 004740 040367 000722      BIC      R3,WORK1      ;MEMORY
1113 004744 000241                CLC
1114 004746 006003                ROR      R3
1115 004750 000770                BR       2$
1116 004752 062767 006250 000706      1$:    ADD      #ENDP,WORK1
1117 004760 016767 000702 000710      MOV      WORK1,BUFF      ;SAVE BUFFER START ADDR.
1118 004766 000207                RTS      PC              ;EXIT
1119                                ;TIMEOUT THE OCCURANCE OF AN INTERRUPT
1120
1121 004770 005000                WAIT:   CLR      RO
1122 004772 005200      2$:    INC      RO
1123 004774 005767 000700                TST      INT              ;HAS INTERRUPT OCCURED?
1124 005000 001005                BNE      1$              ;YES-BRANCH
1125 005002 005700                TST      RO              ;HAS OPERATION TIMED OUT?
1126 005004 001372                BNE      2$              ;NO-BRANCH
1127 005006 104400                HLT
1128 005010 005267 000660                INC      INTERR          ;UNIT TIMED OUT ON READ OR WRITE
1129 005014 000207      1$:    RTS      PC              ;SET ERROR FLAG
1130                                ;EXIT
1131                                ;ENTERED UPON A DEVICE INTERRUPT. THIS ROUTINE
1132                                ;WILL CHECK FOR AND REPORT DEVICE ERRORS
1133
1134 005016 032777 100000 000660      DSKINT: BIT      #B15,DRPCS      ;WHERE THER ANY ERRORS?
1135 005024 001402                BEQ      1$              ;BRANCH-NO ERRORS
1136 005026 000167 000110                JMP      DSKER          ;REPORT ERROR
1137 005032 016703 000640      1$:    MOV      BUFF,R3
1138 005036 062703 001000                ADD      #1000,R3
1139 005042 022765 000004 005460      CMP      #4,DEVTBL(R5)      ;IS THIS A WRITE?
1140 005050 001402                BEQ      3$              ;BRANCH IF YES
1141 005052 062703 001000                ADD      #1000,R3
1142 005056 020377 000630      3$:    CMP      R3,DRPBA      ;DID THE BUS ADDR TERMINATE PROPERLY?
1143 005062 001425                BEQ      2$              ;YES-BRANCH
1144 005064 104400                HLT
1145 005066 004567 174204                JSR      R5,PRINT$      ;PRINT MESSAGE
1146 005072 006125                MES13
1147 005074 004567 174176                JSR      R5,PRINT$      ;PRINT MESSAGE
1148 005100 006155                MES18
1149 005102 010367 174466                MOV      R3,TTY
1150 005106 004767 174244                JSR      PC,PRINTR      ;TYPE LOCATION WITH LEADING ZEROS
1151 005112 004567 174160                JSR      R5,PRINT$      ;PRINT MESSAGE
1152 005116 006171                MES19
1153 005120 017767 000566 174446                MOV      DRPBA,TTY
1154 005126 004767 174224                JSR      PC,PRINTR      ;TYPE LOCATION WITH LEADING ZEROS
1155 005132 005267 000536                INC      INTERR          ;SET ERROR FLAG
1156 005136 000167 000006      2$:    JMP      EXTINT
1157 005142 104400                DSKER:  HLT
1158 005144 005267 000524                INC      INTERR          ;REPORT INTERRUPT DISK ERROR
1159 005150 052767 000001 000522      EXTINT: BIT      #B0,INT      ;SET ERROR FLAG
1160 005156 032777 100000 000536      1$:    BIT      #B15,DRPDS      ;IS THE UNIT READY
1161 005164 001774                BEQ      1$              ;NO-WAIT
1162 005166 000002                RTI
1163

```

```

1164
1165 005170 032767 000002 174056 MSG: BIT #B1,HLTCT$ ;TYPE ENTIRE MSG?
1166 005176 001100 BNE 1$ ;BRANCH IF NO
1167 005200 004567 174072 JSR RS,PRINT$ ;PRINT MESSAGE
1168 005204 005773 MES1
1169 005206 016767 000446 174360 MOV UNIT,TTY
1170 005214 004767 174150 JSR PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
1171 005220 004567 174052 JSR RS,PRINT$ ;PRINT MESSAGE
1172 005224 006025 MES2A
1173 005226 017767 000470 174340 MOV @RPDS,TTY
1174 005234 004767 174116 JSR PC,PRINT$
1175 005240 004567 174032 JSR RS,PRINT$ ;PRINT MESSAGE
1176 005244 006003 MES1A
1177 005246 017767 000446 174320 MOV @RPER,TTY
1178 005254 004767 174076 JSR PC,PRINT$ ;TYPE LOCATION WITH LEADING ZEROS
1179 005260 004567 174012 JSR RS,PRINT$ ;PRINT MESSAGE
1180 005264 006014 MES2
1181 005266 017767 000412 174300 MOV @RPCS,TTY
1182 005274 004767 174056 JSR PC,PRINT$ ;TYPE LOCATION WITH LEADING ZEROS
1183 005300 004567 173772 JSR RS,PRINT$ ;PRINT MESSAGE
1184 005304 006036 MES3
1185 005306 016567 005464 174260 MOV DEVTBL+4(RS),TTY
1186 005314 004767 174050 JSR PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
1187 005320 004567 173752 JSR RS,PRINT$ ;PRINT MESSAGE
1188 005324 006053 MES4
1189 005326 005067 000340 CLR WORK3
1190 005332 116567 005471 000332 MOVB DEVTBL+11(RS),WORK3
1191 005340 016767 000326 174226 MOV WORK3,TTY
1192 005346 004767 174016 JSR PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
1193 005352 004567 173720 JSR RS,PRINT$ ;PRINT MESSAGE
1194 005356 006065 MESS
1195 005360 116567 005470 000304 MOVB DEVTBL+10(RS),WORK3
1196 005366 016767 000300 174200 MOV WORK3,TTY
1197 005374 004767 173770 JSR PC,PRINT$ ;TYPE LOCATION-SUPRESS ZEROS
1198 005400 032767 000001 173646 1$: BIT #B0,HLTCT$ ;TYPE EXPECTED - RECEIVED?
1199 005406 001001 BNE 2$ ;BRANCH IF YES
1200 005410 000207 RTS PC
1201 005412 2$:
1202 005412 004567 173660 JSR RS,PRINT$ ;PRINT MESSAGE
1203 005416 006155 MES18
1204 005420 016767 000030 174146 MOV EXP$,TTY
1205 005426 004767 173724 JSR PC,PRINT$ ;TYPE LOCATION WITH LEADING ZEROS
1206 005432 004567 173640 JSR RS,PRINT$ ;PRINT MESSAGE
1207 005436 006171 MES19
1208 005440 016767 000012 174126 MOV RECS,TTY
1209 005446 004767 173704 JSR PC,PRINT$ ;TYPE LOCATION WITH LEADING ZEROS
1210 005452 000207 RTS PC
1211 005454 000000 EXP$: 0
1212 005456 000000 RECS: 0
1213 ;DEVTBL IS A TABLE CONTAINING SLOTS FOR EACH OF EIGHT
1214 ;POSSIBLE UNITS. DURING THE OPERATION OF THE PROGRAM
1215 ;RS IS USED AS A MODIFIER TO POINT INTO THE TABLE TO
1216 ;SELECT THE PROPER UNIT. EACH UNIT SLOT CONTAINS
1217 ;EIGHT ENTRIES(WORD)
1218 ;1 FUNCTION POINTER
1219 ; 0=SEEK

```

```

1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233 005460 000000
1234      005500
1235 005500 000000
1236      005520
1237 005520 000000
1238      005540
1239 005540 000000
1240      005560
1241 005560 000000
1242      005600
1243 005600 000000
1244      005620
1245 005620 000000
1246      005640
1247 005640 000000
1248      005660
1249
1250
1251 005660 000000
1252 005662 000000
1253 005664 000000
1254 005666 000000
1255 005670 000000
1256 005672 000000
1257 005674 000000
1258 005676 000000
1259 005700 000000
1260 005702 000000
1261
1262 005704 176714
1263 005706 176715
1264 005710 176716
1265 005712 176720
1266 005714 176722
1267 005716 176724
1268 005720 176712
1269 005722 176710
1270 005724 176734
1271      000254
1272      000256
1273
1274
1275

```

```

:      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

DEVTL: 0 .EVEN ;UNIT 0 SLOT
      =DEVTL+20
UNIT1: 0 ;UNIT 1 SLOT
      =UNIT1+20
UNIT2: 0 ;UNIT 2 SLOT
      =UNIT2+20
UNIT3: 0 ;UNIT 3 SLOT
      =UNIT3+20
UNIT4: 0 ;UNIT 4 SLOT
      =UNIT4+20
UNIT5: 0 ;UNIT 5 SLOT
      =UNIT5+20
UNIT6: 0 ;UNIT 6 SLOT
      =UNIT6+20
UNIT7: 0 ;UNIT 7 SLOT
      =UNIT7+20

;RP11 CONSTANTS-MEMORY ASSIGNMENTS
UNIT: 0 ;CURRENT UNIT UNDER TEST
PASSCT: 0 ;COUNTS EACH PASS THRU 8 UNITS
WORK: 0 ;TEMPORARY STORAGE AREA
WORK1: 0
WORK2: 0
WORK3: 0
INTERR: 0 ;INTERRUPT ERROR FLAG
BUFF: 0 ;STARTING ADDRESS OF BUFFER
INT: 0 ;INTERRUPT FLAG
FLAG: 0 ;FLAG WORD

;DISK IO REGISTERS
RPCS: 176714 ;DISK CONTROL REGISTER
RPCS1: 176715
RPWC: 176716 ;DISK WORD COUNT REGISTER
RPBA: 176720 ;CURRENT ADDRESS REGISTER
RPCA: 176722 ;CYLINDER ADDRESS REGISTER
RPDA: 176724 ;DISK ADDRESS REGISTER
RPER: 176712 ;ERROR REGISTER
RPDS: 176710 ;DRIVE STATUS REGISTER
SUCA: 176734 ;CURRENT CYLINDER ADDRESS
VECTOR=254 ;INTERRUPT VECTOR ADDR.
STATUS=256 ;DISK INTERRUPT STATUS

:MESSAGES

```

```

1276 005726 005015 042115 030455 TITLE: .ASCIZ <15><12>/MD-11-DZRPC-C, RP11C MULTI DRIVE/<15><12>
1277 005734 026461 055104 050122
1278 005742 026503 026103 051040
1279 005750 030520 041461 046440
1280 005756 046125 044524 042040
1281 005764 044522 042526 005015
1282 005772 000
1283 005773 015 052412 044516 MES1: .ASCIZ <15><12>/UNIT /
1284 006000 020124 000
1285 006003 015 051012 042520 MES1A: .ASCIZ <15><12>/RPER= /
1286 006010 036522 000040
1287 006014 005015 050122 051503 MES2: .ASCIZ <15><12>/RPCS= /
1288 006022 020075 000
1289 006025 015 051012 042120 MES2A: .ASCIZ <15><12>/RPDS= /
1290 006032 036523 000040
1291 006036 005015 054503 044514 MES3: .ASCIZ <15><12>/CYLINDER= /
1292 006044 042116 051105 020075
1293 006052 000
1294 006053 015 052012 040522 MES4: .ASCIZ <15><12>/TRACK= /
1295 006060 045503 004475 000
1296 006065 015 051412 041505 MES5: .ASCIZ <15><12>/SECTOR= /
1297 006072 047524 036522 000040
1298 006100 005015 040504 040524 MES12: .ASCIZ <15><12>/DATA COMPARE ERROR
1299 006106 041440 046517 040520
1300 006114 042522 042440 051122
1301 006122 051117 000
1302 006125 015 041012 051525 MES13: .ASCIZ <15><12>/BUS ADDRESS INCORRECT
1303 006132 040440 042104 042522
1304 006140 051523 044440 041516
1305 006146 051117 042522 052103
1306 006154 000
1307 006155 015 042412 050130 MES18: .ASCIZ <15><12>/EXPECTED /
1308 006162 041505 042524 020104
1309 006170 000
1310 006171 015 051012 041505 MES19: .ASCIZ <15><12>/RECEIVED /
1311 006176 044505 042526 020104
1312 006204 000
1313 006205 015 042412 042116 MES20: .ASCIZ <15><12>/END QUICK VERIFY PASS/
1314 006212 050440 044525 045503
1315 006220 053040 051105 043111
1316 006226 020131 040520 051523
1317 006234 000
1318 006235 015 042412 042116 MES21: .ASCIZ <15><12>/END PASS/
1319 006242 050040 051501 000123
1320 006250 000000 ENDP: 0 ;START OF BUFFER AREA
1321 006250 000000 .END

```

ATTN	003162	EXTMEN	004554	MES10	003756	PSW	=	177776	T	=	000020					
ATTNB	003160	FLAG	005702	MES11	004005	RANADR	004672		TBITVE	=	000014					
BELL	001262	HINUM	001712	MES12	006100	RANDAT	004410		TITLE	005726						
BUFF	005676	HLT	=	104400	MES13	006125	RANDA1	004434	TKB	=	177562					
BC	=	000001	HLTADS	001256	MES18	006155	RANOS	001576	TKS	=	177560					
B1	=	000002	HLTCTS	001254	MES19	006171	RAND1	004550	TPB	=	177566					
B10	=	002000	HOME	003014	MES2	006014	RAND2	004552	TPS	=	177564					
B11	=	004000	ICNT	001000	MES2A	006025	RDCNT	004402	TRAPVE	=	000034					
B12	=	010000	INPUTS	002026	MES20	006205	READD	004204	TTY	001574						
B13	=	020000	INT	005700	MES21	006235	READL\$	002072	UNIT	005660						
B14	=	040000	INTERR	005674	MES3	006036	READM\$	002066	UNIT1	005500						
B15	=	100000	IOTVEC	=	000020	MES4	006053	READS	001716	UNIT2	005520					
B2	=	000004	JMPTBL	003172	MES5	006065	RECS	005456	UNIT3	005540						
B3	=	000010	KIPAR0	=	172340	MEXIT1	003012	REST\$	001064	UNIT4	005560					
B4	=	000020	KIPAR1	=	172342	MSG	005170	RESVEC	=	000010						
B5	=	000040	KIPAR2	=	172344	N	=	000010	RPBA	005712						
B6	=	000100	KIPAR3	=	172346	NUM\$	002346	RPCA	005714							
B7	=	000200	KIPAR4	=	172350	PACK\$	002074	RPCS	005704							
B8	=	000400	KIPAR5	=	172352	PAC\$	002314	RPCS1	005706							
B9	=	001000	KIPAR6	=	172354	PASSCT	005662	RPDA	005716							
C	=	000001	KIPAR7	=	172356	PFVEC	=	000024	RPDS	005722						
CLRTAB	002446	KIPDR0	=	172300	PKEX\$	002302	RPER	005720	RPWC	005710						
DEVTBL	005460	KIPDR1	=	172302	PKEX1\$	002306	RW	=	000006	SAVE	004670					
DISPLA	=	177570	KIPDR2	=	172304	PK\$	002344	SAVE\$	001036	SCOPE	=	104000				
DSKER	005142	KIPDR3	=	172306	PRINTA	001402	SCOPE\$	001004	SEEK	003202						
DSKINT	005016	KIPDR4	=	172310	PRINTB	001412	SEKCK	003516	SHIFT	004442						
EMTVEC	=	000030	KIPDR5	=	172312	PRINTR	001356	SRO	=	177572						
ENDP	006250	KIPDR6	=	172314	PRINTS	001370	START	002350	STATUS	=	000256					
ERRFLG	001252	KIPDR7	=	172316	PRINT\$	001276	STKPTR	=	000500	SUCA	005724					
ERROR	001112	LAD	001002	PRI1	=	000040	SWR	=	177570							
ERRORS	001260	LONUM	001714	PRI2	=	000100										
ERRPC	001264	LOOP	002654	PRI3	=	000140										
ERRVEC	=	000004	MAIN	002662	PRI4	=	000200									
EXGEN	004546	MAXREF	004632	PRI5	=	000240										
EXPS	005454	MAXRF1	004640	PRI6	=	000300										
EXREF	004604	MES1	005773	PRI7	=	000340										
EXTINT	005150	MES1A	006003	PRTF\$	001314											

. ABS. 006252 000

ERRORS DETECTED: 0

DSKZ: DZRPCC/SOL=DSKZ: SYSMAC.SML, DSKM: DZRPCC.P11  
RUN-TIME: 7 8 .1 SECONDS  
RUN-TIME RATIO: 305/16=18.1  
CORE USED: 33K (65 PAGES)