

RP11C

RELIABILITY DIAGNOSTICS
MD-11-DZRPB-D

EP-DZRPB-D-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

```

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

```

```

000000
000001
000002
000003
000004
000005
000006
000007

```

```

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

```

```

: STATUS REGISTER (PSW) BIT ASSIGNMENTS

```

```

000001
000002
000004
000010
000020
000040
000340
000300
000240
000200
000140
000100
000040

```

```

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

```

```

000004
000010
000014
000020
000024
000030
000034

```

```

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

```

```

: REGISTER ADDRESSES

```

```

177776
177560
177562
177564
177566
177570
177570

```

```

PSW=177776  : PROCESSOR STATUS REGISTER
TKS=177560  : KEYBOARD CSR
TKB=177562  : ADDR OF KEYBOARD BUFFER
TPS=177564  : TELEPRINTER CSR
TPB=177566  : TELEPRINTER BUFFER
SWR=177570  : CONSOLE SWITCH REGISTER
DISPLAY=177570 : CONSOLE DISPLAY REGISTER

```

```

: INITIAL STACK POINTER

```

```

000500

```

```

STKPTR=500

```

```

: PROGRAM STACK POINTER

```

```

: BIT ASSIGNMENTS

```

```

100000

```

```

BIS=100000

```

000000 000001 000002 000003 000004 000005 000006 000007 000010 000020 000040 000340 000300 000240 000200 000140 000100 000040 177776 177560 177562 177564 177566 177570 177570 000500 100000

:INDEX OF CALLS

- SCOPE
- SAVE
- REST
- HLT
- PRINT
- DUMP
- DUMPF
- SCUMP
- SDUMPF
- RAND
- READ
- PACK

13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68

```

000200 000200 002336
001000 001000
001002 000000
001004 032737 040000 177570
001012 001403
001014 005767 000220
001020 001003
001022 005067 000212
001026 000002
001030 016716 177746
001034 000002
001036 012667 000020
001042 C10546

```

```

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

```

```

169 001044 010446      MOV      R4,-(SP)
170 001046 010346      MOV      R3,-(SP)
171 001050 010246      MOV      R2,-(SP)
172 001052 010146      MOV      R1,-(SP)
173 001054 010046      MOV      R0,-(SP)
174 001056 016707 000000    MOV      1$,PC      ;RETURN
175 001062 000000      1$:          0      ;CONTAINS RETURN ADDRESS
;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
;CALLED BY REST MACRO
178 001064 012667 000020    RESTS: MOV      (SP)+,1$      ;SAVE RETURN PC
179 001070 012600      MOV      (SP)+,R0
180 001072 012601      MOV      (SP)+,R1
181 001074 012602      MOV      (SP)+,R2
182 001076 012603      MOV      (SP)+,R3
183 001100 012604      MOV      (SP)+,R4
184 001102 012605      MOV      (SP)+,R5
185 001104 016707 000000    MOV      1$,PC      ;RETURN
186 001110 000000      1$:          0      ;CONTAINS RETURN ADDR
187 ;ERROR SERVICE ROUTINE CALLED BY HLT
188 ;THIS ROUTINE WILL HALT ON ERROR, RING THE BELL, AND
189 ;TRANSFER CONTROL TO A USER SUPPLIED ROUTINE IF SPECIFIED
190 001112 005737 177570    ERROR: TST      2$SWR      ;HALT ON ERROR?
191 001116 100001      BPL      3$          ;BRANCH IF NO
192 001120 000000      HALT
193 001122 032737 004000 177570 3$: BIT      8B11,2$SWR      ;RING THE BELL?
194 001130 001403      BEQ      1$          ;BRANCH IF NO
195 001132 004567 000144      JSR      R5,PRNTFS    ;FORCE PRINT THE MESSAGE
196 001136 001250      BELL
197 001140 032737 020000 177570 1$: BIT      8B13,2$SWR      ;SKIP TYPEOUT?
198 001146 001022      BNE      2$          ;BRANCH IF YES
199 001150 004567 000110      JSR      R5,PRINTS    ;PRINT MESSAGE
200 001154 001252      ERRPC
201 001156 011667 000062      MOV      (6),HLTADS    ;GET ERROR PC+2
202 001162 162767 000002 000054      SUB      2,HLTADS      ;MODIFY
203 001170 117767 000050 000044      MOV      2,HLTADS,HLTCTS ;SAVE HLT ARGUMENT
204 001176 016767 000042 000356      MOV      HLTADS,TTY
205 001204 004767 000134      JSR      PC,PRINTR     ;TYPE LOCATION WITH LEADING ZEROS
206 001210 004767 016102      JSR      PC,MSG        ;GO TO USER ERROR ROUTINE
207 001214 005737 177570      2$: TST      2$SWR      ;HALT ON ERROR?
208 001220 100001      BPL      4$          ;BRANCH IF NO
209 001222 000000      HALT
210 001224 052767 100000 000006 4$: BIS      8B15,ERRFLG    ;SET ERROR FLAG
211 001232 005267 000010      INC      ERRORS        ;UPDATE ERROR COUNTER
212 001236 000002      RTI
213 001240 000000      ERRFLG: 0
214 001242 000000      HLTCTS: 0
215 001244 000000      HLTADS: 0      ;PC OF ERROR
216 001246 000000      ERRORS: 0      ;ERROR COUNT
217 001250 000007      BELL: .ASCIZ <7>
218 001252 005015 005015 041520 ERRPC: .ASCIZ <15><12><15><12>'PC= '
219 001260 020075 000      .EVEN
220 001264
221 ;THIS ROUTINE WILL PRINT AN ASCIZ MESSAGE.
222 ;THE MESSAGE MUST TERMINATE IN 0
223 001264 032737 020000 177570 PRINTS: BIT      8B13,2$SWR      ;INHIBIT TYPEOUTS?
224 001272 001403      BEQ      PRNTFS        ;BRANCH IF NO

```

F01

FRONT END
DZRPB.P11

MACY11 27(732) 04-NOV-76 14:18 PAGE 6

```

225 001274 062705 000002          ADD    #2,R5          ;LDATE RETURN ADDR
226 001300 000205          RTS     R5
227 001302 105737 177564          FRNTFS: TSTB   @#TPS          ;WAIT FOR PRINTER TO FINISH
228 001306 100375          BPL    .-4
229 001310 010546          MOV    R5,-(SP)
230 001312 062716 000002          ADD    #2,(SP)          ;ADJUST RETURN PC
231 001316 011505          MOV    (R5),R5          ;GET MESSAGE ADDR
232 001320 105715          1S:   TSTB   (R5)          ;CHECK FOR TERMINATOR
233 001322 001002          BNE    2S
234 001324 012605          MOV    (SP)+,R5          ;GET RETURN ADDR
235 001326 000205          RTS     R5          ;RETURN
236 001330 112537 177566          2S:   MOVB   (R5)+,@#TPB          ;PRINT CHARACTER
237 001334 105737 177564          TSTB   @#TPS          ;WAIT TILL DONE
238 001340 100375          BPL    .-4
239 001342 000766          BR     1S
240
241 001344 032737 020000 177570          ;THIS ROUTINE TYPES A LOCATION IN CCTAL
242 001352 001406          PRINTR: BIT    #B13,@#SWR          ;INHIBIT TYPEOUT?
243 001354 000207          BEQ    PRINTA          ;BRANCH IF NO
244 001356 032737 020000 177570          PRINTS: BIT    #B13,@#SWR          ;INHIBIT TYPEOUT?
245 001364 001405          BEQ    PRINTB          ;BRANCH IF NO
246 001366 000207          RTS     PC
247 001370 112767 000001 000140          PRINTA: MOVB   #1,.PR          ;SET ZERO FILL SWITCH
248 001376 000402          BR     .+6          ;SKIP
249 001400 005067 000132          PRINTB: CLR    .PR          ;SUPPRESS LEADING ZEROS
250 001404 112767 177772 000125          MOVB   #-6,.PR+1          ;SET COUNT
251 001412 010446          .PTIT: MOV    R4,-(SP)          ;SAVE R4
252 001414 012704 001540          MOV    #.PR+2,R4          ;SET POINTER TO FIRST CHARACTER
253 001420 105014          CLRB   (R4)          ;CLEAR FIRST BYTE
254 001422 000413          BR     .PRF          ;ROTATE FIRST BIT
255 001424 105014          .PRL:  CLRB   (R4)          ;CLEAR BYTE OF CHAR
256 001426 032767 000100 000102          BIT    #100,.PR          ;BIT TYPING MODE
257 001434 001006          BNE    .PRF          ;YES SKIP 2 ROTATES
258 001436 006167 000120          ROL    TTY          ;ROTATE BIT INTO C
259 001442 106114          ROLB   (4)          ;PACK IT
260 001444 006167 000112          ROL    TTY
261 001450 106114          ROLB   (4)
262 001452 006167 000104          .PRF:  ROL    TTY
263 001456 106114          ROLB   (4)
264 001460 105714          TSTB   (4)          ;IS IT ZERO
265 001462 001402          BEQ    .+6          ;SKIP INC
266 001464 105267 000046          INCB   .PR          ;SET FILL SWITCH
267 001470 105767 000042          TSTB   .PR          ;CHECK FILL SWITCH
268 001474 001402          BEQ    .+6          ;SKIP BITSET
269 001476 152724 000060          BISB   #'0,(4)+          ;MAKE INTO ASCIZ CHAR
270 001502 105267 000031          INCB   .PR+1          ;INC COUNT
271 001506 001346          BNE    .PRL          ;REPEAT
272 001510 022704 001540          CMP    #.PR+2,R4          ;EMPTY BUFFER
273 001514 001002          BNE    .+6          ;SKIP IF NOT
274 001516 112724 000060          MOVB   #'0,(4)+          ;LOAD ONE ZERO
275 001522 105014          CLRB   (4)          ;NULL TERMINATOR
276 001524 004567 177534          JSR    R5,PRINTS          ;PRINT MESSAGE
277 001530 001540          .PR+2
278 001532 012604          MOV    (SP)+,R4          ;RESTORE R4
279 001534 000207          RTS     PC
280 001536 000012          .PR:   .BLKW  12

```

281	001562	000000		TTY:	0		
282	001564			RANDS:			
283	001564	004767	177246		JSR	PC, SAVES	;SAVE THE REGISTERS
284	001570	016700	000106		MOV	LONUM, R0	;SET R0 WITH LOW
285	001574	016701	000100		MOV	HINUM, R1	;SET R1 WITH HIGH
286	001600	012703	177771		MOV	#-7, R3	;SET SHIFT COUNT
287	001604	005002			CLR	R2	
288	001606	006300		1S:	ASL	R0	;SHIFT R0 LEFT AND
289	001610	006101			ROL	R1	;ROTATE CARRY INTO R1 AND
290	001612	005102			ROL	R2	;ROTATE CARRY INTO R2
291	001614	005203			INC	R3	;CHECK FOR DONE
292	001616	001373			BNE	1S	
293	001620	066702	000056		ADD	LONUM, R2	;ADD # TO MAKE X 129
294	001624	005501			ADC	R1	;PROPOGATE CARRY
295	001626	066701	000046		ADD	HINUM, R1	;ADD # TO MAKE X 129
296	001632	005502			ADC	R2	;PROPOGATE CARRY
297	001634	062700	001057		ADD	#1057, R0	
298	001640	005501			ADC	R1	;PROPOGATE CARRY
299	001642	005502			ADC	R2	;PROPOGATE CARRY
300	001644	062701	047401		ADD	#47401, R1	
301	001650	005502			ADC	R2	
302	001652	062702	000006		ADD	#6, R2	
303	001656	060200			ADD	R2, R0	
304	001660	005501			ADC	R1	
305	001662	010067	000014		MOV	R0, LONUM	
306	001666	010167	000006		MOV	R1, HINUM	
307	001672	004767	177166		JSR	PC, RESTS	;RESTORE THE REGISTERS
308	001676	000207			RTS	PC	
309							
310	001700	000000		HINUM:	0		
311	001702	000000		LONUM:	0		
312	001704	010346		READS:	MOV	R3, -(6)	;SAVE R3
313	001706	012703	002014	1S:	MOV	#INPUTS, R3	;GET BUFFER ADDR
314	001712	022703	002034	2S:	CMP	#INPUTS+20, R3	;BUFFER FULL?
315	001716	001412			BEQ	4S	;YES..TYPE ?
316	001720	105737	177560		TSTB	@#177560	;WAIT FOR A CHAR
317	001724	100375			BPL	.-4	
318	001726	113713	177562		MOVW	@#177562, (3)	;GET CHAR
319	001732	142713	000200		BICB	#200, (3)	;GET RID OF JUNK
320	001736	122713	000177		CMPB	#177, (3)	;IS IT A RUBOUT?
321	001742	001004			BNE	3S	;SKIP IF NO
322	001744			4S:			
323	001744	004567	177314		JSR	R5, PRINTS	;PRINT MESSAGE
324	001750	002054			READMS		
325	001752	000755			BR	1S	;CLEAR BUFFER AND START OVER
326	001754	013737	177562	177566	3S:	MOV	@#TKB, @#TPB
327	001762	105737	177564		TSTB	@#TPS	;ECHO THE CHAR
328	001766	100375			BPL	.-4	;WAIT FOR READY
329	001770	122723	000015		CMPB	#15, (3)+	;CHECK FOR RETURN
330	001774	001346			BNE	2S	;LOOP IF NOT RETURN
331	001776	105063	177777		CLRB	-1(3)	;REMOVE THE RETURN
332	002002	004567	177256		JSR	R5, PRINTS	;PRINT MESSAGE
333	002006	002060			READLS		
334	002010	012603			MOV	(6)+, R3	;RESTORE R3
335	002012	000207			RTS	PC	;RETURN
336							

H01

FRONT END MACY11 27(732) 04-NOV-76 14:18 PAGE 8
 DZRPB.P11

337	002014	000020			INPUTS: .BLKW 20
338	002054	006477	000012		READMS: .ASCIZ '??' <15> <12>
339	002060	000012			READLS: .ASCIZ <12>
340					
341					:TAKE THE CONTENTS OF THE TTY INPUT BUFFER AND
342					:PACK THEM INTO ONE WORD TO CREATE AN OCTAL NUMBER
343					
344	002062				PACKS:
345	002062	004767	176750		JSR PC,SAVES ;SAVE THE REGISTERS
346	002066	005067	000242		CLR NUMS
347	002072	005000			CLR RO
348	002074	105760	002014	2S:	TSTB INPUTS(RO)
349	002100	001402			BEQ 1S
350	002102	005200			INC RO
351	002104	000773			BR 2S
352	002106	005300		1S:	DEC RO
353	002110	004767	000166		JSR PC,PACS ;GET OCTAL CHAR
354	002114	016767	000212	000212	MOV PKS,NUMS ;PACK FIRST CHAR
355	002122	004767	000154		JSR PC,PACS ;GET OCTAL CHAR
356	002126	000241			CLC
357	002130	006167	000176		ROL PKS
358	002134	006167	000172		ROL PKS
359	002140	006167	000166		ROL PKS
360	002144	056767	000162	000162	BIS PKS,NUMS ;PACK SECOND CHAR
361	002152	004767	000124		JSR PC,PACS ;GET OCTAL CHAR
362	002156	000241			CLC
363	002160	000367	000146		SWAB PKS
364	002164	006067	000142		ROR PKS
365	002170	006067	000136		ROR PKS
366	002174	056767	000132	000132	BIS PKS,NUMS ;PACK THIRD CHAR
367	002202	004767	000074		JSR PC,PACS ;GET OCTAL CHAR
368	002206	000367	000120		SWAB PKS
369	002212	000241			CLC
370	002214	006167	000112		ROL PKS
371	002220	056767	000106	000106	BIS PKS,NUMS ;PACK FOURTH CHAR
372	002226	004767	000050		JSR PC,PACS ;GET OCTAL CHAR
373	002232	000367	000074		SWAB PKS
374	002236	000241			CLC
375	002240	006167	000066		ROL PKS
376	002244	006167	000062		ROL PKS
377	002250	006167	000056		ROL PKS
378	002254	006167	000052		ROL PKS
379	002260	056767	000046	000046	BIS PKS,NUMS ;PACK FIFTH CHAR
380	002266	000402			BR PKEX1S
381	002270	062706	000002		ADD #2,SP ;MODIFY STACK
382	002274				PKEX1S:
383	002274	004767	176564		JSR PC,RESTS ;RESTORE THE REGISTERS
384	002300	000207			RTS ;EXIT
385					
386	002302	005700			PACS: TST RO
387	002304	100771			BMI PKEXS
388	002306	005067	000020		CLR PKS
389	002312	116067	002014	000012	MOVB INPUTS(RO),PKS ;GET INPUT CHAR
390	002320	005300			DEC RO
391	002322	042767	177770	000002	BIC #177770,PKS ;CLEAR UNWANTED BITS
392	002330	000207			RTS PC

FRONT END
DZRF8.P11

MRCY11 27(732) 04-NOV-76 14:18 PAGE 9

393		
394	002332	000000
395	002334	000000
396		

PKS: 0
NUMS: 0

.TITLE RP11C RELIABILITY TEST

```

397
398
399
400
401 002336 000005          START: RESET          ;CLEAR THE WORLD
402 002340 012706 000500      MOV      #STKPTR,SP    ;SETUP STACK
403 002344 004767 014272      JSR      PC,INIT      ;INITIALIZE VECTORS
404 002350 004567 176726      JSR      RS,PRNTFS    ;FORCE PRINT THE MESSAGE
405 002354 025726          HEADER
406 002356 005067 023250      CLR      FLAG         ;CLEAR PROGRAM FLAG
407 002362 032737 000100 177570 BIT      #B6,@#SWR    ;WANT TO USE MEMORY MANAGEMENT?
408 002370 001034          BNE      2$           ;BR IF NO
409 002372 012737 002462 000004 MOV      #2$,@#ERRVEC ;SETUP TRAP TEST FOR MEMORY MANAGEMENT
410 002400 012737 000340 000006 MOV      #340,@#ERRVEC+2
411 002406 005737 177572      TST      @#SRO        ;MEMORY MANAGEMENT?
412 002412 005037 172340      CLR      @#KIPAR0     ;YES! SET UP TO USE MEMORY MANAGEMENT
413 002416 012737 000200 172342 MOV      #200,@#KIPAR1 ;SECOND 4K PAGE
414 002424 012737 007600 172356 MOV      #7600,@#KIPAR7 ;I/O PAGE
415 002432 012737 177406 172300 MOV      #400*256.-400+UP+RW,@#KIPDR0 ;SET KIPDR0=RW UP 400 BLOCKS
416 002440 012737 177406 172302 MOV      #400*256.-400+UP+RW,@#KIPDR1 ;SET KIPDR1=RW UP 400 BLOCKS
417 002446 012737 177406 172316 MOV      #400*256.-400+UP+RW,@#KIPDR7 ;SET KIPDR7=RW UP 400 BLOCKS
418 002454 005237 177572      INC      @#SRO        ;TURN ON MEMORY MANAGEMENT
419 002460 000403          BR       3$
420 002462 052767 000100 023142 2$: BIS      #B6,FLAG     ;BIT6 SET = NO MEMORY MANAGEMENT
421 002470 012737 000006 000004 3$: MOV      #ERRVEC+2,@#ERRVEC ;RESTORE TRAP CATCHER
422 002476 005037 000006          CLR      @#ERRVEC+2
423 002502 005067 023220      CLR      HEADER      ;ASCII TERMINATOR SO ON RESTART THERE IS NO HEADER
424 002506 004567 016206      JSR      RS,EXTMEN    ;SET UP DATA BUFFERS
425 002512 005067 023202      CLR      SEEK        ;INITIALIZE SEEK RANDOM NUMBER GENERATOR
426 002516 005067 176256      CLR      ICNT        ;CLEAR THE PASS COUNTER
427 002522 005067 023134      CLR      DSKNOR      ;CLEAR UNIT FLAG
428 002526 005067 023112      CLR      CYLINDER    ;CLEAR THE CYLINDER ADDRESS
429 002532 005067 023110      CLR      DMA         ;CLEAR DAR REGISTERS
430 002536 005067 023110      CLR      PATNU       ;CLEAR PATTERN COUNT
431 002542 005067 023154      CLR      CNTA        ;CLEAR READ COUNTER FOR DATA AND RANDOM TEST
432 002546 032737 000400 177570 BIT      #B8,@#SWR    ;USE CONVERSATION MODE?
433 002554 001005          BNE      LCONM       ;BRANCH IF YES
434 002556 052767 070000 023046 BIS      #70000,FLAG
435 002564 000167 000756          JMP      ADTST
436          ;ENTER OPERATOR CONVERSATION MODE
437 002570 013746 177570      LCONM: MOV      @#SWR,-(SP) ;PUT SWR ONTO STACK
438 002574 042716 177774      BIC      #177774,(SP) ;CLEAN OUT UNWANTED SWITCHES
439 002600 022726 000003      CMP      #3,(SP)+    ;TEST 3?
440 002604 001402          BEQ      1$         ;BR IF YES
441 002606 000167 017666      JMP      CYLSK       ;GO DO ADDRESS CONVERSATION
442          1$:
443 002612 004567 176464      JSR      RS,PRNTFS    ;FORCE PRINT THE MESSAGE
444 002616 024501          SPECMES ;STANDARD WORDS TRANSFERED =
445 002620 016767 023032 176734 MOV      SWRDCT,TTY
446 002626 004767 176546      JSR      PC,PRINTB   ;FORCE TYPE LOCATION - SUPPRESS ZEROS
447 002632 004567 176444      JSR      RS,PRNTFS    ;FORCE PRINT THE MESSAGE
448 002636 024530          CON1  ;ASK ABOUT DATA TEST ONLY
449 002640 004767 177040      JSR      PC,READS     ;INPUT MESSAGE
450 002644 122767 000131 177142 CMPB     #131,INPUTS  ;TEST FOR YES
451 002652 001003          BNE      .+10        ;BRANCH IF NO
452 002654 052767 002000 022750 BIS      #B10,FLAG    ;SET DATA TEST ONLY FLAG

```

453	002662	004567	176414		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
454	002666	024563			CON2			; ASK ABOUT MULTI DRIVE MODE
455	002670	004767	177010		JSR	PC,READS		; INPUT MESSAGE
456	002674	122767	000131	177112	CMPB	#13!,INPUTS		; TEST FOR YES
457	002702	001040			BNE	DATTES		; BRANCH IF NO
458	002704	052767	004000	022720	BIS	#B11,FLAG		; SET MULTI UNIT FLAG
459	002712							
460	002712	004567	176364		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
461	002716	024517			CON3			; GET NO. OF UNITS
462	002720	004767	176760		JSR	PC,READS		; INPUT MESSAGE
463	002724	004767	177132		JSR	PC,PACKS		; CONVERT INPUT TO A NUMBER
464	002730	005767	177400		TST	NUMS		; IS IT ZERO
465	002734	001766			BEQ	DSKDR		
466	002736	162767	000001	177370	SUB	#1,NUMS		
467	002744	022767	000010	177362	CMP	#10,NUMS		; IS NO. TOO HIGH
468	002752	101757			BLOS	DSKDR		
469	002754	016767	177354	022700	MOV	NUMS,DSKNOR		; SAVE HIGHEST UNIT NO.
470	002762	042767	177770	022672	BIC	#177770,DSKNOR		
471	002770	000241			CLC			
472	002772	006167	022664		ROL	DSKNOR		
473	002776	006167	022660		ROL	DSKNOR		
474	003002	000423			BR	ASKWC		
475	003004							
476	003004	004567	176272		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
477	003010	024561			CON4			; ASK UNIT NUMBER
478	003012	004767	176666		JSR	PC,READS		; INPUT MESSAGE
479	003016	004767	177040		JSR	PC,PACKS		; CONVERT INPUT TO A NUMBER
480	003022	022767	000010	177304	CMP	#10,NUMS		; IS NO = OR>10
481	003030	101765			BLOS	DATTES		; NO
482	003032	000241			CLC			
483	003034	006167	177274		ROL	NUMS		
484	003040	006167	177270		ROL	NUMS		
485	003044	056767	177264	022560	BIS	NUMS,FLAG		; SAVE UNIT UNDER TEST
486	003052							
487	003052	004567	176224		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
488	003056	024700			CON5			; ASK ABOUT OPTIONAL WORD COUNT
489	003060	004767	176620		JSR	PC,READS		; INPUT MESSAGE
490	003064	122767	000131	176722	CMPB	#13!,INPUTS		; TEST FOR YES
491	003072	001034			BNE	TKSR		; ASK ABOUT OPTIONAL DAR
492	003074							
493	003074	004567	176202		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
494	003100	024740			CON6			; ASK LENGTH OF WC
495	003102	004767	176576		JSR	PC,READS		; INPUT MESSAGE
496	003106	004767	176750		JSR	PC,PACKS		; CONVERT INPUT TO A NUMBER
497	003112	005767	177216		TST	NUMS		
498	003116	001766			BEQ	WCCON		
499	003120	016767	022532	022554	MOV	SWRDC,WORK		
500	003126	005267	022550		INC	WORK		
501	003132	026767	022544	177174	CMP	WORK,NUMS		; IS NO. GREATER THAN AVAILABLE CORE?
502	003140	101755			BLOS	WCCON		; YES ASK FOR COUNT AGAIN
503	003142	016767	177166	022506	MOV	NUMS,SWRDC		; OPERATING WORD COUNT
504	003150	016767	022502	022464	MOV	SWRDC,WRDCT		
505	003156	052767	000002	022446	BIS	#B1,FLAG		; OPERATOR SELECTED WORD COUNT
506	003164							
507	003164	004567	176112		JSR	RS,PRNTFS		; FORCE PRINT THE MESSAGE
508	003170	025011			CON7A			; ASK ABOUT DISK ADDR

509	003172	004767	176506		JSR	PC, READS	; INPUT MESSAGE
510	003176	122767	000131	176610	CMPB	#131, INPUTS	; WILL OPERATOR SUPPLY ADDR?
511	003204	001055			BNE	OPPAT	; BRANCH IF NO
512	003206	052767	000040	022416	BIS	#B5, FLAG	
513	003214				OPDAR:		
514	003214	004567	176062		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
515	003220	025116			CON7C		; GET CYLINDER ADDR
516	003222	004767	176456		JSR	PC, READS	; INPUT MESSAGE
517	003226	004767	176630		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
518	003232	022767	000626	177074	CMP	#626, NUMS	; IS CYLINDER LEGAL
519	003240	101765			BLOS	OPLAR	
520	003242	016767	177066	022364	MOV	NUMS, SCYL	; SAVE ADDR
521	003250				OPDA1:		
522	003250	004567	176026		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
523	003254	024770			CON7		; GET HEAD ADDR
524	003256	004767	176422		JSR	PC, READS	; INPUT MESSAGE
525	003262	004767	176574		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
526	003266	022767	000024	177040	CMP	#24, NUMS	
527	003274	101765			BLOS	OPDA1	; BRANCH IF HEAD ADDR TOO HIGH
528	003276	016767	177032	022332	MOV	NUMS, SHED	; SAVE ADDR
529	003304				OPDA2:		
530	003304	004567	175772		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
531	003310	025075			CON7B		; GET SECTOR ADDR
532	003312	004767	176366		JSR	PC, READS	; INPUT MESSAGE
533	003316	004767	176540		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
534	003322	022767	000012	177004	CMP	#12, NUMS	; IS SECTOR ADDR TOO HIGH?
535	003330	101765			BLOS	OPDA2	
536	003332	016767	176776	022300	MOV	NUMS, SSEC	; SAVE ADDR
537							
538	003340				OPPAT:		
539	003340	004567	175736		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
540	003344	025136			CONB		; ASK ABOUT DATA PATTERNS
541	003346	004767	176332		JSR	PC, READS	; INPUT MESSAGE
542	003352	004767	176504		JSR	PC, PACKS	; CONVERT INPUT TO A NUMBER
543	003356	022767	000020	176750	CMP	#20, NUMS	; TEST FOR CORRECT NO
544	003364	101765			BLOS	OPPAT	; ASK AGAIN
545	003366	022767	000017	176740	CMP	#17, NUMS	
546	003374	001411			BEQ	OPWRT	; DATA PATTERN UNDER PROGRAM CONTROL
547	003376	052767	100000	022226	BIS	#B15, FLAG	; SET PROGRAM FLAG
548	003404	016767	176724	022240	MOV	NUMS, PATNU	; OPERATOR WANTS TO SELECT DATA
549	003412	000241			CLC		
550	003414	006167	022232		ROL	PATNU	
551	003420	042767	070000	022204	OPWRT:	BIC	#70000, FLAG
552	003426	004567	175650		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
553	003432	025164			CON9		; ASK ABOUT WRITE
554	003434	004767	176244		JSR	PC, READS	; INPUT MESSAGE
555	003440	122767	000131	176346	CMPB	#131, INPUTS	; TEST FOR YES
556	003446	001003			BNE	OPRD	; ASK ABOUT WRITE CHECK
557	003450	052767	040000	022154	BIS	#B14, FLAG	; YES SET FLAG BIT
558	003456				OPRD:		
559	003456	004567	175620		JSR	RS, PRNTFS	; FORCE PRINT THE MESSAGE
560	003462	025234			CON11		; ASK ABOUT READ
561	003464	004767	176214		JSR	PC, READS	; INPUT MESSAGE
562	003470	122767	000131	176316	CMPB	#131, INPUTS	; TEST FOR YES ANSWER
563	003476	001003			BNE	OPWCK	
564	003500	052767	010000	022124	BIS	#B12, FLAG	; SET FLAG TO READ

565	003506					OPWCK:			
566	003506	004567	175570			JSR	R5, PRNTFS		; FORCE PRINT THE MESSAGE
567	003512	025205				CON10			; ASK ABOUT WRITE CHECK
568	003514	004767	176164			JSR	PC, READS		; INPUT MESSAGE
569	003520	122767	000131	176266		CMPB	#131, INPUTS		
570	003526	001003				BNE	CHKMOD		
571	003530	052767	020000	022074		BIS	#B13, FLAG		; SET WRITE CHECK FLAG
572	003536	032767	070000	022066		CHKMOD:	#70000, FLAG		; MAKE SURE SOME OPERATION WAS SELECTED
573	003544	001725				BEQ	OPWRT		
574									
575									
576	003546	005737	000042			ADTST:	TST	#42	; UNDER MONITOR CONTROL?
577	003552	001444					BEQ	1\$; BRANCH IF NO
578	003554	005067	022102				CLR	DSKNOR	
579	003560	012777	000001	022020	3\$:	MOV	#1, DRPCS		; CLEAR THE RP11C
580	003566	116777	022070	022014		MOVB	DSKNOR, DRPCS1		; SELECT THE DRIVE
581	003574	005777	022026			TST	DRPDS		; IS THE UNIT READY?
582	003600	100003				BPL	2\$; BRANCH IF NO
583	003602	005267	022054			INC	DSKNOR		; UPDATE UNIT NUMBER
584	003606	000764				BR	3\$		
585	003610	005367	022046		2\$:	DEC	DSKNOR		; DSKNOR = NUMBER OF UNITS
586	003614	000241				CLC			
587	003616	006167	022040			ROL	DSKNOR		
588	003622	006167	022034			ROL	DSKNOR		
589	003626	052767	004000	021776		BIS	#B11, FLAG		; SET MULTI DRIVE FLAG
590	003634	005767	022022			TST	DSKNOR		; WERE ANY UNITS AVAILABLE?
591	003640	100011				BPL	1\$; BRANCH IF YES
592	003642	004567	175416			JSR	R5, PRINTS		; PRINT MESSAGE
593	003646	024207				MES20			
594	003650	013701	000042			MOV	#42, R1		; ABORT - NO UNITS AVAILABLE
595	003654	005067	174162			CLR	42		; SET ABORT FLAG
596	003660	000167	010614			JMP	MEXIT		
597	003664	032767	004000	021740	1\$:	BIT	#B11, FLAG		; ARE WE IN MULTI DRIVE MODE?
598	003672	001422				BEQ	EXMFLG		; BRANCH IF NO
599	003674	004567	175402			JSR	R5, PRNTFS		; FORCE PRINT THE MESSAGE
600	003700	024061				MES11			
601	003702	016767	021724	021764		MOV	FLAG, ACNVX		; TELL OPERATOR THE UNIT UNDER TEST
602	003710	006067	021760			ROR	ACNVX		
603	003714	006067	021754			ROR	ACNVX		
604	003720	042767	177770	021746		BIC	#177770, ACNVX		
605	003726	016767	021742	175626		MOV	ACNVX, TTY		
606	003734	004767	175440			JSR	PC, PRINTS		; FORCE TYPE LOCATION - SUPPRESS ZEROS
607	003740	032737	000010	177570	EXMFLG:	BIT	#B3, SWR		; RUN SELECTED TEST?
608	003746	001410				BEQ	1\$; BRANCH IF NO
609	003750	013700	177570			MOV	#SWR, RO		; GET SWITCH SETTINGS
610	003754	042700	177770			BIC	#177770, RO		
611	003760	000241				CLC			
612	003762	006100				ROL	RO		
613	003764	000170	004004			JMP	DTSTTBL(RO)		; GO TO SELECTED TEST
614	003770	032767	002000	021634	1\$:	BIT	#B10, FLAG		; DATA TEST ONLY?
615	003776	001412				BEQ	ADT1		; NO
616	004000	000167	006266			JMP	DATAT		; DO DATA TEST
617									
618	004004	004024				TSTTBL:	ADT1		
619	004006	004502					ADT2		
620	004010	005564					ADT3		

621 004012 006230
622 004014 010526
623 004016 012272
624 004020 013366
625 004022 014514

WRCK
MEMTST
DATAT
RANEX
PFTST

.5bTTL ***** TEST 0 *****

; IN THIS TEST THE PROGRAM SEEKS FROM 0 TO N AND THEN BACK
; TO 0. N STARTS AT ZERO THEN INCREMENTS TO 1 AND UP THRU 625
; DONE IS TIMED OUT, SELECTED UNIT CYLINDER ADDRESS IS TESTED, SEEK UNDERWAY
; IS CHECK, AND THE ATTENTION FLAG IS TESTED.

637 004024 005067 021646
638 004030 004567 175230
639 004034 023740
640 004036 016767 021634 175516
641 004044 004767 175306
642 004050 012777 004470 021524
643 004056 012777 000340 021520
644 004064 004567 012500
645 004070 005067 021550
646 004074 005067 021606
647 004100 005067 021604
648 004104 012737 000200 177776
649 004112 117777 021510 021506
650 004120 016777 021520 021470
651 004126 005067 000346
652 004132 052777 020011 021446
653 004140 012700 000025
654 004144 005300
655 004146 001376
656 004150 105777 021432
657 004154 100402
658 004156 104400
659 004160 000467
660 004162 005767 021520
661 004166 001406
662 004170 032777 002000 021430
663 004176 001002
664 004200 104400
665 004202 000456
666 004204 005000
667 004206 005200
668 004210 005777 021412
669 004214 100414
670 004216 005237 025714
671 004222 005237 025714
672 004226 005337 025714
673 004232 005337 025714
674 004236 005700
675 004240 001362
676 004242 104400

ADT1: CLR TESTNO
JSR RS,PRINT\$;PRINT MESSAGE
MES6
MOV TESTNO,TTY
JSR PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
RADT1: MOV #INTCK,VECTOR ;SET UP DISK VECTOR
MOV #340,STATUS
JSR RS,DSKNOS ;SELECT UNIT
CLR CYLINDER
CLR WORK2 ;CYLINDER COUNTER
CLR WORK3 ;POINTER
MOV #PRI4,PSW ;ALLOW INTERRUPTS
9\$: MOVB ARPDS,ARPDS ;CLEAR ATTENTION BITS
MOV CYLINDER,ARPCA ;SET CYLINDER REGISTER
CLR INTFLG ;CLEAR INTERRUPT FLAG
BIS #20011,ARPCS ;SEEK AND ENABLE ATTN INTERRUPT
1\$: MOV #25,RO
DEC RO ;DELAY FOR DONE TO SET
BNE 1\$
TSTB ARPCS ;TEST FOR DONE
BMI 2\$;BRANCH DONE SET
HLT ;DONE DID NOT SET AFTER SEEK
2\$: TST WORK2 ;DON'T TEST SEEK UNDERWAY
BEQ 3\$;IF FIRST TIME THRU
EIT #B10,ARPDS ;DID SEEK UNDERWAY SET?
BNE 3\$;BRANCH IF YES
HLT ;SEEK UNDERWAY DID NOT SET
3\$: CLR RO
5\$: INC RO ;TIMEOUT UNIT READY
TST ARPDS ;IS UNIT READY?
BMI 6\$;BRANCH IF YES
INC #CYLA
INC #CYLA
DEC #CYLA
DEC #CYLA
TST RO ;TIMEOUT?
BNE 5\$;BRANCH IF NO
HLT ;READY DID NOT SET AFTER SEEK

700	000000	000000	000000	000000	65:	MOV	01,INTFLG	: DID INTERRUPT OCCUR?
701	000000	000000	000000	000000		INTFLG	125	: BRANCH IF YES
702	000000	000000	000000	000000				: INTERRUPT DID NOT OCCUR ON ATTENTION BIT
703	000000	000000	000000	000000	125:	TST	ATTN	: DETERMINE ATTENTION BIT
704	000000	000000	000000	000000		ATTN	2,ARPC	: IS ATTENTION BIT SET?
705	000000	000000	000000	000000				: BRANCH IF YES
706	000000	000000	000000	000000				: ATTENTION BIT DID NOT SET
707	000000	000000	000000	000000	75:	MOV	01,SLCA	: IS SLCA CORRECT?
708	000000	000000	000000	000000		SLCA	2,SLCA	: EXPECTED RESULTS
709	000000	000000	000000	000000		SLCA	2,SLCA	: RECEIVED RESULTS
710	000000	000000	000000	000000				: CONTENTS OF SLCA INCORRECT
711	000000	000000	000000	000000	115:	TST	0,ARPC	: ANY DEVICE ERRORS
712	000000	000000	000000	000000				: BRANCH IF NO
713	000000	000000	000000	000000				: DEVICE ERROR AFTER SEEK OPERATION
714	000000	000000	000000	000000	85:	BIT	0B11,ARPC	: SEEK INCOMPLETE ERROR?
715	000000	000000	000000	000000				: BRANCH IF NO
716	000000	000000	000000	000000		MOV	015,ARPC	: ISSUE HOME COMMAND
717	000000	000000	000000	000000		TST	0,ARPC	: WAIT FOR DONE
718	000000	000000	000000	000000	135:	BIT	0B15,ARPC	: WAIT FOR UNIT READY
719	000000	000000	000000	000000				
720	000000	000000	000000	000000	45:	MOV	095,LAD	: SET UP LOOP
721	000000	000000	000000	000000		SCOPE		
722	000000	000000	000000	000000		TST	WORK3	: SEEK CYLINDER ZERO?
						BMI	105	: BRANCH IF YES
						INC	WORK2	: UPDATE CYLINDER
						MOV	WORK2,CYLINDER	
						BIS	0B15,WORK3	: SET SEEK ZERO FLAG
					105:	BR	95	
							CYLINDER	
							WORK3	: CLEAR SEEK ZERO FLAG
							0625,WORK2	: HAS LAST CYLINDER BEEN REACHED?
							95	: BRANCH IF NO
						BIT	0B10,05SWR	: REPEAT TEST
						BEQ	ADT2	: NO-GO TO NEXT
						JMP	ADT1	: YES
720	004470	012767	000001	000002	INTCK:	MOV	01,INTFLG	: SET INTERRUPT FLAG
721	004476	000002				RTI		
722	004500	000000			INTFLG:	0		

732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778

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

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

```

004502 012767 000001 021166 ADT2:  MOV    #1,TESTNO
004510 004567 174550          JSR    RS,PRINTS      ;PRINT MESSAGE
004514 023740          MES6
004516 016767 021154 175036  MOV    TESTNO,TTY
004524 004767 174626          JSR    PC,PRINTS      ;TYPE LOCATION-SUPPRESS ZEROS
004530 005067 015216          CLR    MEX            ;CLEAR DRIVE EXTENDED MEMORY BITS
004534 004567 012030  RADT2: JSR    RS,DSKNOS     ;SELECT THE DRIVE
004540 052777 000015 021040 15:    BIS    #15,DRPCS     ;SEEK HOME
004546 012700 000025          MOV    #25,RO
004552 005300          25:    DEC    RO            ;GIVE DONE A CHANCE TO SET
004554 001376          BNE    25
004556 105777 021024          TSTB   DRPCS         ;IS DONE SET?
004562 100402          BMI    35            ;YES-BRANCH
004564 104400          HLT
004566 000425          BR     65            ;DONE DID NOT SET AFTER A SEEK HOME
004570 005000          35:    CLR    RO            ;CHECK FOR LOOPING
004572 005200          55:    INC    RO
004574 005777 021026          TST    DRPDS         ;IS UNIT READY?
004600 100414          BMI    45            ;YES BRANCH
004602 005237 025714          INC    #CYLA
004606 005237 025714          INC    #CYLA
004612 005337 025714          DEC    #CYLA
004616 005337 025714          DEC    #CYLA
004622 005700          TST    RO            ;HAS UNIT TIMED OUT
004624 001362          BNE    55            ;NO-BRANCH
004626 104400          HLT
004630 000404          BR     65            ;READY DID NOT SET AFTER HOME SEEK
004632 005777 020750          45:    TST    DRPCS         ;CHECK FOR LOOPING
004636 100001          BPL    65            ;ANY ERRORS?
004640 104400          HLT
004642 012767 004540 174132 65:    MOV    #15,LAD      ;NO-BRANCH
004650 104000          SCOPE
004652 012767 005000 020762  MOV    #5000,WRDCT   ;DRIVE ERRORS AFTER HOME SEEK
004660 012767 025730 020766  MOV    #OUTBUF,BUF   ;SETUP LOOP ADDRESS
004666 005067 020754          CLR    DMA
004672 005067 020746          CLR    CYLINDER
004676 012700 025730          SEABUF: MOV    #OUTBUF,RO   ;GET BUFFER STARTING ADDR
004702 012701 000400 215:  MOV    #400,R1      ;SECTOR COUNT
004706 016720 020732          MOV    CYLINDER,(RO)+ ;GENERATE PATTERN SO THAT THE
004712 005301          DEC    R1            ;THE FIRST WORD OF EACH SECTOR
004714 016720 020726 15:    MOV    DMA,(RO)+    ;EQUALS THE CYLINDER ADDR AND
004720 005301          DEC    R1            ;THE REMAINDER EQUALS THE HEAD AND
004722 001374          BNE    15            ;SECTOR ADDR
  
```


779	004724	122767	000011	020714		CMPB	#11,DMA	
780	004732	001403				BEQ	22\$	
781	004734	005267	020706			INC	DMA	;UPDATE SECTOR COUNT
782	004740	000760				BR	21\$	
783	004742	105067	020700		22\$:	CLRB	DMA	
784	004746	004567	010162		4\$:	JSR	R5,FUNCT	;WRITE TEN SECTORS
785	004752	000003			.WORD	3		
786	004754	005000				CLR	RO	
787	004756	005200			23\$:	INC	RO	
788	004760	105777	020622			TSTB	2RPCS	;IS DONE SET?
789	004764	100404				BMI	2\$;YES BRANCH
790	004766	005700				TST	RO	;TEST FOR TIMEOUT
791	004770	001372				BNE	3\$;BRANCH IF NO
792	004772	104400				HLT		;DONE DID NOT SET AFTER WRITE
793	004774	000501				BR	5\$	
794	004776	005777	020604		25\$:	TST	2RPCS	;ANY DEVICE ERRORS?
795	005002	100002				BPL	6\$;BRANCH IF NO
796	005004	104400				HLT		;RPIIC STATUS ERROR AFTER WRITE
797	005006	000474				BR	5\$	
798	005010	005067	020666		6\$:	CLR	WORK	;INCREMENT FLAG
799	005014	017767	020600	012736		MOV	2RPDA,RECS	;GET DISK ADDR
800	005022	042767	177760	012730		BIC	#177760,RECS	;SAVE SECTOR ADDR
801	005030	005767	012724			TST	RECS	
802	005034	001404				BEQ	9\$;BRANCH IF SECTOR = ZERO
803	005036	005067	012714			CLR	EXPS	
804	005042	104401				HLT	+1	;SECTOR ADDR IN RPDA DID NOT UPDATE
805	005044	000455				BR	5\$;PROPERLY AFTER A TEN SECTOR WRITE
806	005046	117767	020550	012704	9\$:	MOVB	2RPDA1,RECS	;GET THE HEAD ADDR
807	005054	116767	020567	012674		MOVB	DMA+1,EXPS	;SECTOR ADDR OUTPUTTED
808	005062	122767	000023	012666		CMPB	#23,EXPS	;DID WE OUTPUT HEAD 23?
809	005070	001005				BNE	7\$;BRANCH IF NO
810	005072	005067	012660			CLR	EXPS	;RESET HEAD ADDR
811	005076	010667	020600			MOV	SP,WORK	;SET INCREMENT FLAG
812	005102	000402				BR	8\$	
813	005104	005267	012646		7\$:	INC	EXPS	
814	005110	126767	012642	012642	8\$:	CMPB	EXPS,RECS	;IS DISK HEAD ADDR CORRECT?
815	005116	001402				BEQ	12\$;BRANCH IF YES
816	005120	104401				HLT	+1	;HEAD ADDR IN RPDA WAS INCORRECT
817	005122	000426				BR	5\$;AFTER TEN SECTOR WRITE
818	005124	017767	020466	012626	12\$:	MOV	2RPCA,RECS	;GET DISK CYLINDER ADDR
819	005132	016767	020506	012616		MOV	CYLINDER,EXPS	
820	005140	005767	020536			TST	WORK	;IS INCREMENT FLAG SET?
821	005144	001410				BEQ	13\$;BRANCH IF NO
822	005146	005267	012604			INC	EXPS	
823	005152	022767	000626	012576		CMP	#626,EXPS	;WAS IT LAST CYLINDER?
824	005160	001002				BNE	13\$;BRANCH IF NO
825	005162	005067	012570			CLR	EXPS	
826	005166	026767	012564	012564	13\$:	CMP	EXPS,RECS	;IS DISK CYLINDER ADDR CORRECT?
827	005174	001401				BEQ	5\$;BRANCH IF YES
828	005176	104401				HLT	+1	;CYLINDER ADDR IN RPCA IS NOT
829								;CORRECT AFTER TEN SECTOR WRITE
830	005200	032777	004000	020420	5\$:	BIT	#B11,2RPDS	;SEEK INCOMPLETE ERROR?
831	005206	001412				BEQ	10\$;BRANCH IF NO
832	005210	112777	000015	020370		MOVB	#15,2RPCS	;ISSUE HOME COMMAND
833	005216	105777	020364			TSTB	2RPCS	;WAIT FOR DONE
834	005222	100375				BPL	.-4	

835	005224	032777	100000	020374	11\$:	BIT	#B15,DRPDS	;WAIT FOR UNIT READY
836	005232	001774				BEQ	11\$	
837	005234	012767	004746	173540	10\$:	MOV	#4\$,LAD	;SETUP LOOP ADDR
838	005242	104000				SCOPE		
839	005244	004767	010710			JSR	PC,DISBUF	;SETUP NEXT DISK ADDR
840	005250	000612				BR	SEABUF	;WRITE NEXT SECTOR
841	005252	012767	005000	020362		MOV	#5000,WRDCT	;RESTORE WORD COUNT
842	005260	052777	000015	020320		BIS	#15,DRPCS	;SEEK HOME
843	005266	105777	020314			TSTB	DRPCS	
844	005272	100375				BPL	.-4	;WAIT FOR DONE AFTER SEEK HOME
845	005274	005777	020326			TST	DRPDS	
846	005300	100375				BPL	.-4	;WAIT FOR DRIVE READY AFTER SEEK HOME
847	005302	012700	025730		POSECT:	MOV	#OUTBUF,RO	
848	005306	012701	005000			MOV	#5000,R1	
849	005312	005020			23\$:	CLR	(R0)+	;CLEAR THE BUFFER
850	005314	005301				DEC	R1	
851	005316	001375				BNE	23\$	
852	005320	004567	007610			JSR	R5,FUNCT	;READ TEN SECTORS
853	005324	000005			.WORD	5		
854	005326	105777	020254			TSTB	DRPCS	
855	005332	100375				BPL	.-4	;WAIT FOR DONE AFTER READ
856	005334	005777	020246			TST	DRPCS	;ANY ERRORS?
857	005340	100006				BPL	ADHGT	;BRANCH NO ERRORS
858	005342	104400				HLT		;STATUS ERROR AFTER A READ
859	005344	032777	040000	020234		BIT	#B14,DRPCS	;WAS IT A DATA ERROR?
860	005352	001401				BEQ	ADHGT	;IF YES GO COMPARE DATA
861	005354	000446				BR	ADTER1	
862	005356	012700	025730		ADHGT:	MOV	#OUTBUF,RO	
863	005362	012701	000400		ADHGT1:	MOV	#400,R1	
864	005366	026710	020252			CMP	CYLINDER,(0)	;IS CYLINDER WORD CORRECT?
865	005372	001017				BNE	ADERC	;BRANCH IF NO
866	005374	005720				TST	(0)+	
867	005376	005301				DEC	R1	
868	005400	026710	020242		SANHT:	CMP	DMA,(0)	;IS HEAD-SECTOR WORD CORRECT?
869	005404	001016				BNE	ADERR	;BRANCH IF NO
870	005406	005720				TST	(0)+	
871	005410	005301				DEC	R1	
872	005412	001372				BNE	SANHT	
873	005414	122767	000011	020224		CMPB	#11,DMA	
874	005422	001023				BEQ	ADTER1	
875	005424	005267	020216			INC	DMA	
876	005430	000754				BR	ADHGT1	
877	005432	016767	020206	012316	ADERC:	MOV	CYLINDER,EXPS	;CORRECT DATA/ADDRESS
878	005440	000403				BR	ADERC1	
879	005442	016767	020200	012306	ADERR:	MOV	DMA,EXPS	;CORRECT DATA/ADDRESS
880	005450	011067	012304		ADERC1:	MOV	(0),RECS	;INCORRECT DATA
881	005454	104401				HLT	+1	;DATA COMPARE ERROR
882	005456	022701	000400			CMP	#400,R1	;WAS FIRST WORD INCORRECT?
883	005462	001003				BNE	ADTER1	;BRANCH IF NO
884	005464	004567	173574			JSR	R5,PRINT\$;PRINT MESSAGE
885	005470	024075				MES12		;WRONG CYLINDER PROBABLY SELECTED
886	005472	105067	020150		ADTER1:	CLRB	DMA	
887	005476	032777	004000	020122		BIT	#B11,DRPDS	;SEEK INCOMPLETE ERROR?
888	005504	001412				BEQ	1\$;BRANCH IF NO
889	005506	112777	000015	020072		MOV#B	#15,DRPCS	;ISSUE HOME COMMAND
890	005514	105777	020066			TSTB	DRPCS	;WAIT FOR DONE

891	005520	100375				BPL	.-4	
892	005522	032777	100000	020076	2S:	BIT	#B15,DRPDS	;WAIT FOR UNIT READY
893	005530	001774				BEQ	2S	
894	005532	012767	005302	173242	1S:	MOV	#RDSECT,LAD	;SETUP LOOP
895	005540	104000				SCOPE		
896	005542	004767	010412			JSR	PC,DISBUF	;SETUP NEXT DISK ADDRESS
897	005546	000655				BR	RDSECT	;CHECK NEXT SECTOR
898	005550	032737	002000	177570		BIT	#B10,DRSWR	;LOOP ON TEST?
899	005556	001402				BEQ	ADT3	;BRANCH IF NO
900	005560	000167	176750			JMP	RADT2	

```

901          .SBTTL ***** TEST 2 *****
902
903          ;WRITE THE FIRST WORD OF EACH CYLINDER WITH THE CYLINDER ADDRESS.
904          ;THEN SEEK FROM 625 TO 0. THEN SEEK TO 624 AND BACK TO 1
905          ;UNTIL THE NUMBERS CROSS. AFTER EACH SEEK VERIFY THE POSITION
906          ;BY READING THE FIRST WORD ON THE CYLINDER AND COMPARING
907          ;AGAINST CYLINDER REQUESTED.
908
909 005564 012767 000002 020104 ADT3:  MOV    #2,TESTNO
910 005572 004567 173466          JSR    R5,PRINT$      ;PRINT MESSAGE
911 005576 023740          MESS6
912 005600 016767 020072 173754  MOV    TESTNO,TTY
913 005606 004767 173544          JSR    PC,PRINTS     ;TYPE LOCATION-SUPPRESS ZEROS
914 005612 005067 014134          CLR    MEX           ;CLEAR DRIVE EXTENDED MEMORY BITS
915 005616 004767 011020  RADT3: JSR    PC,INIT      ;INITIALIZE VECTORS
916 005622 004567 010742          JSR    R5,DSKNOS    ;SELECT UNIT
917 005626 005067 020014          CLR    DMA
918 005632 005067 020006          CLR    CYLINDER
919 005636 012767 000001 017776  MOV    #1,WRDCT     ;SET UP WORD COUNT
920 005644 012767 025730 020002  MOV    #OUTBUF,BUF  ;SETUP BUFFER ADDR
921 005652 016737 017766 025730  WRCYL: MOV    CYLINDER,#OUTBUF ;INSERT PATTERN
922 005660 004567 007250          JSR    R5,FUNCT     ;WRITE PATTERN ON FIRST SECTOR
923 005664 000003          .WORD 3             ;OF CYLINDER
924 005666 105777 017714          TSTB   @RPCS        ;WAIT FOR DONE
925 005672 100375          BPL    -4
926 005674 005777 017706          TST   @RPCS        ;AND ERRORS?
927 005700 100002          BPL    1$          ;BRANCH IF NO
928 005702 104400          HLT
929 005704 000407          BR    2$          ;DEVICE ERROR WHILE WRITING
930 005706 022767 000625 017730  1$:  CMP    #625,CYLINDER ;ALL CYLINDERS WRITTEN?
931 005714 001403          BEQ    2$          ;BRANCH IF YES
932 005716 005267 017722          INC    CYLINDER
933 005722 000753          BR    WRCYL
934 005724 032777 004000 017674  2$:  BIT    #B11,@RPS   ;SEEK INCOMPLETE ERROR?
935 005732 001412          BEQ    3$          ;BRANCH IF NO
936 005734 112777 000015 017644  MOVB   #15,@RPCS   ;ISSUE HOME COMMAND
937 005742 105777 017640          TSTB   @RPCS        ;WAIT FOR DONE
938 005746 100375          BPL    -4
939 005750 032777 100000 017650  4$:  BIT    #B15,@RPS   ;WAIT FOR UNIT READY
940 005756 001774          BEQ    4$
941 005760 012767 005652 173014  3$:  MOV    #WRCYL,LAD  ;SETUP UP LOOP
942 005766 104000          SCOPE
943 005770 005067 017650          CLR    CYLINDER
944 005774 012767 000624 017704  MOV    #624,WORK2
945 006002 005067 017702          CLR    WORK3
946 006006 005067 017700          CLR    WORK4
947 006012 016767 017652 017634  ADT32: MOV    INBUF,BUF    ;INC - DEC FLAG
948 006020 004567 007110          JSR    R5,FUNCT     ;READ THE FIRST WORD OF THE
949 006024 000005          .WORD 5             ;CYLINDER
950 006026 105777 017554          TSTB   @RPCS        ;WAIT FOR DONE AFTER READ
951 006032 100375          BPL    -4
952 006034 005777 017546          TST   @RPCS        ;ANY ERRORS?
953 006040 100002          BPL    3$          ;BRANCH IF NO
954 006042 104400          HLT
955 006044 000413          BR    5$
956 006046 027767 017616 017570  3$:  CMP    @INBUF,CYLINDER ;COMPARE DATA READ AGAINST CYLINDER

```

957	006054	001407				BEQ	55		; BRANCH IF EQUAL
958	006056	016767	017562	011672		MOV	CYLINDER,EXPS		; CORRECT DATA
959	006064	017767	017600	011666		MOV	QINBUF,RECS		; INCORRECT DATA
960	006072	104401				HLT	+1		; DATA COMPARE ERROR-PROBABLY WENT
961									; TO THE WRONG CYLINDER
962	006074	032777	004000	017524	55:	BIT	#B11,QRPDS		; SEEK INCOMPLETE ERROR?
963	006102	001412				BEQ	25		; BRANCH IF NO
964	006104	112777	000015	017474		MOVB	#15,QRPCS		; ISSUE A HOME COMMAND
965	006112	105777	017470			TSTB	QRPCS		; WAIT FOR DONE
966	006116	100375				BPL	.-4		
967	006120	032777	100000	017500	45:	BIT	#B15,QRPDS		; WAIT FOR UNIT READY
968	006126	001774				BEQ	45		
969	006130	012767	006012	172644	25:	MOV	#ADT32,LAD		; SETUP LOOP
970	006136	104000				SCOPE			
971	006140	005767	017546			TST	WORK4		; INC - DEC FLAG
972	006144	100411				BMI	15		
973	006146	005267	017536			INC	WORK3		; UPDATE LOW COUNT
974	006152	016767	017530	017464		MOV	WORK2,CYLINDER		
975	006160	052767	100000	017524		BIS	#B15,WORK4		; SET DECREMENT FLAG
976	006166	000711				BR	ADT32		
977	006170	005367	017512		15:	DEC	WORK2		; DECREMENT HIGH COUNT
978	006174	005067	017512			CLR	WORK4		; CLEAR FLAG
979	006200	016767	017504	017436		MOV	WORK3,CYLINDER		
980	006206	005767	017474			TST	WORK2		; DONE YET
981	006212	001277				BNE	ADT32		; BRANCH-NO
982	006214	032737	002000	177570		BIT	#B10,Q#SWR		; LOOP ON TEST?
983	006222	001402				BEQ	WRCK		; NO
984	006224	000167	177366			JMP	RADT3		; YES

```

985
986
987
988
989
990
991
992
993
994
995
996 006230 012767 000003 017440 WRCK:  MOV    #3,TESTNO
997 006236 004567 173022          JSR    R5,PRINT$      ;PRINT MESSAGE
998 006242 023740          MES6
999 006244 016767 017426 173310  MOV    TESTNO,TTY
1000 006252 004767 173100          JSR    PC,PRINT$     ;TYPE LOCATION-SUPPRESS ZEROS
1001 006256 005067 013470          CLR    MEX           ;CLEAR EXTENDED MEMORY BITS IN CONTROLER
1002 006262 004767 010354          RWRCK: JSR    PC,INIT   ;INITIALIZE
1003 006266 004567 010276          JSR    R5,DSKNOS    ;SELECT UNIT
1004 006272 005067 017346          CLR    CYLINDER
1005 006276 005067 017344          CLR    DMA
1006 006302 005000          CLR    RO           ;PATTERN FLAG
1007 006304 012701 000001          MOV    #1,R1        ;STARTING PATTERN
1008 006310 012767 025730 017336  MOV    #OUTBUF,BUF  ;SETUP OUTPUT BUFFER
1009 006316 012767 000400 017316  MOV    #400,WRDCT   ;SETUP WORDCOUNT
1010 006324 005002          23$:  CLR    R2
1011 006326 010162 025730          1$:  MOV    R1,OUTBUF(R2) ;GENERATE TEST PATTERN
1012 006332 005722          TST    (R2)+        ;UPDATE MODIFIER
1013 006334 022702 000400          CMP    #400,R2     ;HAS BUFFER BEEN FILLED?
1014 006340 001372          BNE    1$          ;BRANCH IF NO
1015 006342 004567 006566          JSR    R5,FUNCT    ;WRITE PATTERN
1016 006346 000003          .WORD 3
1017 006350 105777 017232          TSTB   @RPCS       ;WAIT FOR DONE
1018 006354 100375          BPL    -4
1019 006356 005777 017224          TST    @RPCS       ;ANY DEVICE ERRORS?
1020 006362 100002          BPL    2$          ;BRANCH IF NO
1021 006364 104400          HLT
1022 006366 000475          BR     3$          ;ERROR AFTER WRITING ONE SECTOR
1023 006370 004567 006540          2$:  JSR    R5,FUNCT    ;WRITE CHECK THE DATA
1024 006374 000007          .WORD 7
1025 006376 105777 017204          TSTB   @RPCS       ;WAIT FOR DONE
1026 006402 100375          BPL    -4
1027 006404 005777 017176          TST    @RPCS       ;ANY DEVICE ERRORS?
1028 006410 100012          BPL    4$          ;BRANCH IF NO
1029 006412 104400          HLT
1030 006414 004567 172644          JSR    R5,PRINT$   ;PRINT MESSAGE
1031 006420 024042          MES10
1032 006422 016767 017302 173132  MOV    OUTBUF,TTY
1033 006430 004767 172710          JSR    PC,PRINT$   ;TYPE LOCATION WITH LEADING ZEROS
1034 006434 000452          BR     3$
1035 006436 005700          4$:  TST    RO           ;ARE WE FLOATING A ONE?
1036 006440 001411          BEQ    20$         ;BRANCH IF YES
1037 006442 005002          CLR    R2          ;FILL BUFFER WITH ONES
1038 006444 012762 177777 025730  21$:  MOV    #177777,OUTBUF(R2)
1039 006452 005722          TST    (R2)+
1040 006454 022702 000400          CMP    #400,R2

```

1041	006460	001371			BNE	21\$		
1042	006462	000407			BR	22\$		
1043	006464	005002		20\$:	CLR	R2		
1044	006466	005062	025730	5\$:	CLR	OUTBUF(R2)	;CLEAR OUTPUT BUFFER	
1045	006472	005722			TST	(R2)+		
1046	006474	022702	000400		CMP	#400,R2	;ENTIRE BUFFER CLEAR?	
1047	006500	001372			BNE	5\$;BRANCH IF NO	
1048	006502	004567	006426	22\$:	JSR	R5,FUNCT	;WRITE CHECK THE DATA AND	
1049	006506	000007		.WORD	7		;EXPECT AN ERROR	
1050	006510	105777	017072		TSTB	2RPCS	;WAIT FOR DONE	
1051	006514	100375			BPL	.-4		
1052	006516	032777	000010	017100	BIT	#B3,2RPER	;IS WRITE CHECK ERROR SET?	
1053	006524	001011			BNE	6\$;BRANCH IF YES	
1054	006526	104400			HLT		;WRITE CHECK ERROR DID NOT SET	
1055	006530	004567	172530		JSR	R5,PRINT\$;PRINT MESSAGE	
1056	006534	024042			MES10			
1057	006536	010167	173020		MOV	R1,TTY		
1058	006542	004767	172576		JSR	PC,PRINTR	;TYPE LOCATION WITH LEADING ZEROS	
1059	006546	000405			BR	3\$		
1060	006550	005777	017032	6\$:	TST	2RPCS	;DID ERROR FLAG SET?	
1061	006554	100402			BMI	3\$;BRANCH IF YES	
1062	006556	104400			HLT		;ERROR FLAG DID NOT SET AFTER WRITE CHECK ERROR	
1063	006560	000400			BR	3\$		
1064	006562	012767	006324	172212	3\$:	MOV	#23\$,LAD	;SETUP LOOP ADDR
1065	006570	104000			SCOPE			
1066	006572	005700			TST	R0	;ARE WE FLOATING A ONE?	
1067	006574	001013			BNE	9\$;BRANCH IF NO	
1068	006576	000241			CLC			
1069	006600	006101			ROL	R1	;ROTATE PATTERN	
1070	006602	103402			BCS	10\$;BRANCH IF COMPLETE	
1071	006604	000167	177514		JMP	23\$		
1072	006610	012700	000001	10\$:	MOV	#1,R0	;SET PATTERN FLAG	
1073	006614	012701	077777		MOV	#077777,R1	;SET NEW PATTERN IN R1	
1074	006620	000167	177500		JMP	23\$		
1075	006624	000241		9\$:	CLC			
1076	006626	006201			ASR	R1	;ROTATE FLOATING ZERO PATTERN	
1077	006630	052701	100000		BIS	#B15,R1		
1078	006634	103002			BCC	PATFIL	;HAS ZERO BEEN FLOATED	
1079	006636	000167	177462		JMP	23\$;JUMP IF NO	
1080								
1081								
1082								
1083								
1084								
1085								
1086								
1087								
1088								
1089	006642	012701	025730		PATFIL: MOV	#OUTBUF,R1		
1090	006646	012700	177777		MOV	#177777,R0		
1091	006652	012702	000400		MOV	#400,R2		
1092	006656	010021		1\$:	MOV	R0,(R1)+	;GENERATE ALL ONES PATTERN	
1093	006660	005302			DEC	R2		
1094	006662	001375			BNE	1\$		
1095	006664	004567	006244		JSR	R5,FUNCT	;WRITE SECTOR WITH ONES	
1096	006670	000003		.WORD	3			

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

```

1097 006672 105777 016710      TSTB  @RPCS      ;WAIT FOR DONE
1098 006676 100375      BPL   -4
1099 006700 005777 016702      TST   @RPCS      ;ANY DEVICE ERRORS
1100 006704 100002      BPL   @S         ;BRANCH IF NO
1101 006706 104400      HLT
1102 006710 000473      BR    @S         ;ERROR AFTER WRITING ONE SECTOR ALL 1'S
1103 006712 012767 000002 016722 2$:  MOV   #2,WRDCT   ;SETUP FOR TWO WORD WRITE
1104 006720 004567 006210      JSR   R5,FUNCT   ;WRITE TWO WORD
1105 006724 000003      .WORD 3
1106 006726 105777 016654      TSTB  @RPCS      ;WAIT FOR DONE
1107 006732 100375      BPL   -4
1108 006734 005777 016646      TST   @RPCS      ;ANY ERRORS?
1109 006740 100002      BPL   @S         ;BRANCH IF NO
1110 006742 104400      HLT
1111 006744 000455      BR    @S         ;ERROR ON ONE WORD WRITE
1112 006746 012767 000400 016666 4$:  MOV   #400,WRDCT ;SETUP WORD COUNT
1113 006754 004567 006154      JSR   R5,FUNCT   ;READ SECTOR
1114 006760 000005      .WORD 5
1115 006762 105777 016620      TSTB  @RPCS      ;WAIT FOR DONE
1116 006766 100375      BPL   -4
1117 006770 005777 016612      TST   @RPCS      ;ANY ERRORS
1118 006774 100006      BPL   @S         ;BRANCH IF NO
1119 006776 104400      HLT
1120 007000 032777 040000 016600      BIT   #814,@RPCS ;WAS IT A DATA ERROR?
1121 007006 001401      BEQ   @S         ;BRANCH IF YES
1122 007010 000433      BR    @S
1123 007012 022767 177777 016710 5$:  CMP   #177777,OUTBUF ;COMPARE FIRST WORD SHOULD BE ONES
1124 007020 001410      BEQ   @S         ;BRANCH IF OK
1125 007022 012767 177777 010726      MOV   #177777,EXPS
1126 007030 016767 016674 010722      MOV   OUTBUF,RECS
1127 007036 104401      HLT   +1         ;DATA COMPARE ERROR ON FIRST
1128 007040 000417      BR    @S         ;WORD READ
1129 007042 012700 025734      MOV   #OUTBUF+4,R0 6$:
1130 007046 012701 000374      MOV   #374,R1
1131 007052 005720      TST   (R0)+      8$:
1132 007054 001003      BNE   @S         ;REMAINDER OF SECTOR SHOULD BE CLEAR
1133 007056 005301      DEC   R1         ;BRANCH IF NO
1134 007060 001374      BNE   @S
1135 007062 000406      BR    @S
1136 007064 016067 177776 010666 7$:  MOV   -2(R0),RECS
1137 007072 005067 010660      CLR   EXPS
1138 007076 104401      HLT   +1         ;DATA FOUND IN AREA OF SECTOR
1139                                     ;WHICH SHOULD HAVE BEEN CLEARED
1140                                     ;BY A ONE WORD WRITE
1141 007100 012767 006642 171674 3$:  MOV   #PATFIL,LAD  ;SET UP LOOP ADDR
1142 007106 104000      SCOPE
1143
1144
1145 ;CHECK THE SETTING OF EOP WHEN TRYING TO WRITE BEYOND
1146 ;THE LIMITS OF THE PACK. THE FIRST SECTOR OF THE PACK IS
1147 ;WRITTEN WITH ZEROS. THEN A TWO SECTOR WRITE OF ALL
1148 ;ONE'S IS ISSUED FOR CYLINDER 625, HEAD 23, AND SECTOR 11.
1149 ;EOP AND ERROR BITS SHOULD SET. THE FIRST SECTOR OF THE
1150 ;PACK IS CHECKED TO MAKE SURE IT IS STILL ZERO.
1151
1152 007110 005067 016532      EOPTST: CLR DMA ;CLEAR DISK ADDRESS

```


1153	007114	005067	016524		CLR	CYLINDER	
1154	007120	012767	000400	016514	MOV	#400,WRDCT	;SET WORDCOUNT TO ONE SECTOR
1155	007126	012767	025730	016520	MOV	#OUTBUF,BUF	;SETUP OUTPUT BUFFER
1156	007134	005001			CLR	R1	
1157	007136	005061	025730		1\$: CLR	OUTBUF(R1)	;CLEAR THE OUTPUT BUFFER
1158	007142	005721			TST	(R1)+	
1159	007144	022701	000400		CMP	#400,R1	
1160	007150	001372			BNE	1\$	
1161	007152	004567	005755		JSR	R5,FUNCT	;WRITE SECTOR ZERO WITH ZEROS
1162	007156	000003		.WORD	3		
1163	007160	105777	016422		TSTB	0RPCS	;WAIT FOR DONE
1164	007164	100375			BPL	.-4	
1165	007166	005777	016414		TST	0RPCS	;ANY DEVICE ERRORS
1166	007172	100002			BPL	2\$;BRANCH IF NO
1167	007174	104400			HLT		;ERROR AFTER WRITING SECTOR ZERO WITH ZEROS
1168	007176	000502			BR	3\$	
1169	007200	012767	001000	016434	2\$: MOV	#1000,WRDCT	;SET WORDCOUNT EQUAL TO TWO SECTORS
1170	007206	012767	000625	016430	MOV	#625,CYLINDER	;SELECT CYLINDER 625
1171	007214	012767	000011	015424	MOV	#11,DMA	;SELECT RECTOR 11
1172	007222	112767	000023	016417	MOV	#23,DMA+1	;SELECT HEAD 23
1173	007230	012702	177777		MOV	#177777,R2	
1174	007234	005001			CLR	R1	
1175	007236	010261	025730		4\$: MOV	R2,OUTBUF(R1)	;SET OUTPUT BUFFER TO ONES
1176	007242	005721			TST	(R1)+	
1177	007244	022701	001000		CMP	#1000,R1	
1178	007250	001372			BNE	4\$	
1179	007252	004567	005656		JSR	R5,FUNCT	;ISSUE TWO SECTOR WRITE TO
1180	007256	000003		.WORD	3		;CYLINDER 625, HEAD 23, AND SECTOR 11
1181	007260	105777	016322		TSTB	0RPCS	;WAIT FOR DONE
1182	007264	100375			BPL	.-4	
1183	007266	032777	000002	016330	BIT	#B1,0RPER	;DID EOP ERROR FLAG SET?
1184	007274	001002			BNE	5\$;BRANCH IF SET
1185	007276	104400			HLT		;EOP ERROR FLAG DID NOT SET WHEN
1186	007300	000441			BR	3\$;WRITE OPERATOR EXCEEDS THE PACK
1187	007302	032777	100000	016276	5\$: BIT	#B15,0RPCS	;DID THE ERROR FLAG SET?
1188	007310	001002			BNE	6\$;BRANCH IF SET
1189	007312	104400			HLT		;ERROR DID NOT SET AFTER GENERATING
1190	007314	000433			BR	3\$;EOP
1191	007316	012767	000002	016316	6\$: MOV	#2,WRDCT	
1192	007324	005067	016316		CLR	DMA	CLEAR THE DISK ADDRESS
1193	007330	005067	016310		CLR	CYLINDER	
1194	007334	004567	005574		JSR	R5,FUNCT	;READ THE FIRST SECTOR OF THE PACK
1195	007340	000005		.WORD	5		;AND EXPECT TO FIND ZEROS
1196	007342	105777	016240		TSTB	0RPCS	;WAIT FOR READY
1197	007346	100375			BPL	.-4	
1198	007350	005777	016232		TST	0RPCS	;WERE THERE ANY ERRORS?
1199	007354	100002			BPL	7\$;BRANCH IF NO
1200	007356	104400			HLT		;ERROR ENCOUNTERED ON 2 WORD READ
1201	007360	000411			BR	3\$;OF FIRST SECTOR ON THE PACK
1202	007362	016767	016342	010370	7\$: MOV	OUTBUF,RECS	;SET FIRST WORD OF BUFFER
1203	007370	005767	010364		TST	RECS	;DOES 1ST SECTOR STILL CONTAIN ZEROS?
1204	007374	001403			BEQ	3\$;BRANCH IF YES
1205	007376	005067	010354		CLR	EXPS	
1206	007402	104401			HLT	+1	;CONTENTS OF THE FIRST SECTOR OF THE
1207							;PACK CHANGED AFTER FORCING EOP
1208							;ERROR, OPERATION PROBABLY

```

1209                                     ;WRAPPED AROUND.
1210 007404 012767 007110 171370 3$:  MOV  #EOPTST,LAD
1211 007412 104000                                     SCOPE
1212
1213
1214                                     ;CHECK THE ABILITY OF THE RP11C TO GENERATE A
1215                                     ;PROGRAM ERROR IF A COMMAND IS ISSUED WHILE THE
1216                                     ;CONTROLLER IS BUSY.
1217
1218 007414 012777 000001 016164 LOKOUT: MOV  #1,ARPCS      ;CLEAR THE CONTROLLER
1219 007422 005067 016216         CLR  CYLINDER      ;CLEAR THE DISK ADDRESS
1220 007426 005067 016214         CLR  DMA
1221 007432 012767 002000 016202     MOV  #2000,WRDCT    ;SETUP WORDCOUNT
1222 007440 004567 005470         JSR  R5,FUNCT     ;ISSUE A READ COMMAND TO GET THE
1223 007444 000005         .WORD 5                ;CONTROLLER BUSY
1224 007446 012700 000100         MOV  #100,R0
1225 007452 005300         1$:  DEC  R0                ;ALLOW THE OPERATION TO PROCEED AWHILE
1226 007454 001376         BNE  1$
1227 007456 112777 000005 016122     MOVB #5,ARPCS     ;ISSUE READ COMMAND WHILE BUSY
1228 007464 105777 016116         TSTB ARPCS       ;WAIT FOR DONE
1229 007470 100375         BPL  -4
1230 007472 032777 002000 016124     BIT  #B10,ARPER  ;DID PROGRAM ERROR SET?
1231 007500 001002         BNE  2$                ;BRANCH IF SET
1232 007502 104400         HLT                                     ;PROGRAM ERROR DID NOT SET WHEN A
1233                                     ;READ COMMAND WAS ISSUED WHILE
1234                                     ;THE DEVICE WAS BUSY
1235 007504 000405
1236 007506 032777 100000 016072 2$:  BR   3$
1237 007514 001001         BIT  #B15,ARPCS  ;DID THE ERROR FLAG SET?
1238 007516 104400         BNE  3$                ;BRANCH IF SET
1239                                     ;ERROR FLAG DID NOT SET AFTER
1240 007520 012767 007414 171254 3$:  MOV  #LOKOUT,LAD ;PROGRAM ERROR
1241 007526 104000                                     SCOPE
1242
1243
1244                                     ;UNFORMAT THE FIRST SECTOR ON THE PACK. THEN READ IT BACK
1245                                     ;AND VERIFY THAT READ AND WRITE HEADER OPERATIONS WILL
1246                                     ;TRANSFER DATA CORRECTLY. NOW THAT THE HEADER IS MIS FORMATTED,
1247                                     ;ISSUE A WRITE COMMAND TO SECTOR ZERO. THIS SHOULD RESULT
1248                                     ;IN SETTING HEADER NOT FOUND. THEN REFORMAT SECTOR ZERO.
1249
1250 007530 012777 000001 016050 HNFCK: MOV  #1,ARPCS      ;CLEAR THE CONTROLLER
1251 007536 005067 016102         CLR  CYLINDER      ;CLEAR DISK ADDR
1252 007542 005067 016100         CLR  DMA
1253 007546 012767 025730 016100     MOV  #OUTBUF,BUF  ;SETUP BUFFER ADDR
1254 007554 012700 025730         MOV  #OUTBUF,R0
1255 007560 012720 000001         MOV  #1,(R0)+
1256 007564 012720 000001         MOV  #1,(R0)+
1257 007570 012720 000001         MOV  #1,(R0)+
1258 007574 012767 000003 016040     MOV  #3,WRDCT    ;LOAD WORDCOUNT
1259 007602 004567 005326         JSR  R5,FUNCT     ;ISSUE WRITE HEADER COMMAND
1260 007606 014003         .WORD 14003     ;TO MISFORMAT SECTOR ZERO
1261 007610 105777 015772         TSTB ARPCS       ;WAIT FOR READY
1262 007614 100375         BPL  -4
1263 007616 005777 015764         TST  ARPCS       ;ANY ERRORS?
1264 007622 100002         BPL  1$                ;BRANCH IF NO

```



```

1303 010066 012767 007740 170706 3S:  MOV      #HNF1,LAD
1304 010074 104000          SCOPE
1305 010076 005000          HNF2:  CLR      RO
1306 010100 005060 025730 1S:  CLR      CUTBUF(RO) ;CLEAR BUFFER
1307 010104 062700 000002          ADD      #2,RO ;UPDATE MODIFIER
1308 010110 022700 000006          CMP      #5,RO ;ENTIRE BUFFER CLEARED?
1309 010114 001371          BNE      1S ;BRANCH IF NO
1310 010116 012767 000003 015516          MOV      #3,WORDCT ;LOAD WORD COUNT
1311 010124 004567 005004          JSR      RS,FUNCT ;ISSUE WRITE HEADER COMMAND
1312 010130 074003          .WORD 14003
1313 010132 105777 015450          TSTB    @RPCS ;WAIT FOR READY
1314 010136 100375          BPL     -4
1315 010140 005777 015442          TST     @RPCS ;ANY ERRORS?
1316 010144 100001          BPL     2S
1317 010146 104400          HLT
1318 010150 012767 010076 170624 2S:  MOV      #HNF2,LAD
1319 010156 104000          SCOPE

;ISSUE A SEEK COMMAND AND WAIT FOR DONE TO SET. THEN ISSUE A WRITE
;COMMAND WHILE THE HEADS ARE STILL MOVING. THE RPIIC SHOULD
;HOLD THE WRITE COMMAND TILL THE SEEK IS COMPLETE.

1320 010160 012777 000001 015420 SKTST: MOV      #1,@RPCS ;CLEAR THE CONTROLLER
1321 010166 004567 006376          JSR      RS,DSKNOS ;SELECT THE UNIT
1322 010172 005077 015420          CLR      @RPCA ;CLEAR THE CYLINDER ADDR REGISTER
1323 010176 005077 015416          CLR      @RPDA ;CLEAR THE DISK ADDR REGISTER
1324 010202 112777 000011 015376          MOVB    #11,@RPCS ;ISSUE A SEEK TO ZERO
1325 010210 105777 015372          TSTB    @RPCS ;WAIT FOR DONE
1326 010214 100375          BPL     -4
1327 010216 032777 100000 015402 1S:  BIT     #B15,@RPDS ;WAIT FOR UNIT READY
1328 010224 001774          BEQ     1S
1329 010226 012777 000300 015362          MOV      #300,@RPCA ;SELECT CYLINDER 300
1330 010234 112777 000011 015344          MOVB    #11,@RPCS ;ISSUE SEEK TO CYLINDER 300
1331 010242 105777 015340          TSTB    @RPCS ;WAIT FOR DONE
1332 010246 100375          BPL     -4
1333 010250 012777 177777 015334          MOV      #-1,@RPWC ;SETUP WORD COUNT
1334 010256 112777 000003 015322          MOVB    #3,@RPCS ;ISSUE A WRITE WHILE HEADS ARE MOVING
1335 010264 105777 015316          TSTB    @RPCS ;WAIT FOR DONE
1336 010270 100375          BPL     -4
1337 010272 005777 015310          TST     @RPCS ;ANY ERRORS?
1338 010276 100001          BPL     2S ;BRANCH IF NO
1339 010300 104400          HLT ;ERROR FOUND AFTER ISSUING A WRITE
;COMMAND WHILE THE HEADS ARE STILL
;MOVING AFTER A SEEK.
;SEEK INCOMPLETE ERROR?
;BRANCH IF NO
;ISSUE A HOME COMMAND
;WAIT FOR DONE

1340 010302 032777 004000 015316 2S:  BIT     #B11,@RPDS ;WAIT FOR UNIT READY
1341 010310 001412          BEQ     3S
1342 010312 112777 000015 015266          MOVB    #15,@RPCS ;BRANCH IF NO
1343 010320 105777 015262          TSTB    @RPCS ;ISSUE A HOME COMMAND
1344 010324 100375          BPL     -4 ;WAIT FOR DONE
1345 010326 032777 100000 015272 4S:  BIT     #B15,@RPDS ;WAIT FOR UNIT READY
1346 010334 001774          BEQ     4S
1347 010336 012767 010160 170436 3S:  MOV      #SKTST,LAD
1348 010344 104000          SCOPE

;CHECK THE ABILITY OF CLEAR TO TERMINATE AN OPERATION
;AND SET READY.

```

1377										
1378	010346	004567	006216		CLRTST:	JSR	R5,CSKNOS		:	SELECT THE UNIT
1379	010352	005067	015270			CLR	DMA		:	CLEAR DISK ADDR
1380	010356	005067	015262			CLR	CYLINDER			
1381	010362	012767	000001	015252		MOV	#1,WRDCT		:	SETUP WORD COUNT
1382	010370	012767	025730	015256		MOV	#OUTBUF,BUF		:	SETUP BUFFER ADDR
1383	010376	004567	004532			JSR	R5,FUNCT		:	ISSUE A WRITE COMMAND
1384	010402	000003			.WORD	3				
1385	010404	012760	000100			MOV	#100,R0			
1386	010410	005300			1S:	DEC	R0		:	WAIT A WHILE THEN ISSUE A
1387	010412	001376				BNE	1S		:	CLEAR COMMAND
1388	010414	112777	000001	015164		MOVB	#1,DRPCS		:	CLEAR THE CONTROLLER
1389	010422	105777	015160			TSTB	DRPCS		:	IS READY SET?
1390	010426	100401				BMI	2S		:	BRANCH IF YES
1391	010430	104400				HLT			:	READY DID NOT SET AFTER ISSUING
1392									:	A CLEAR COMMAND DURING A WRITE.
1393	010432	000005			2S:	RESET				
1394	010434	012767	010346	170340		MOV	#CLRTST,LAD			
1395	010442	104000				SCOPE				
1396	010444	016767	015162	015230		MOV	FLAG,WORK		:	GET UNIT NUMBER
1397	010452	000241				CLC				
1398	010454	006067	015222			ROR	WORK			
1399	010460	006067	015216			ROR	WORK			
1400	010464	000367	015212			ROR	WORK			
1401	010470	042767	174377	015204		SWAB	WORK		:	JUSTIFY UNIT NUMBER
1402	010476	016777	015200	015102		BIC	#174377,WORK		:	CLEAR UNWANTED BITS
1403	010504	005777	015116			MOV	WORK,DRPCS		:	LOAD UNIT NUMBER
1404	010510	100375				TST	DRPCS		:	WAIT FOR UNIT READY
1405	010512	032737	002000	177570		BPL	-4			
1406	010520	001402				BIT	#B10,2#SWR		:	LOOP ON TEST?
1407	010522	000167	175534			BEQ	MEMTST		:	BRANCH IF NO
						JMP	RWRCK			

1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463

.SBTTL ***** TEST 4 *****

: THIS ROUTINE CONSIST OF TWO SEGMENTS. THE FIRST
: PART TEST THE ACCESSIBILITY OF MEMORY WITHOUT
: UTILIZING MEMORY MANAGEMENT. EACH LOCATION
: FROM THE END OF THE PROGRAM TO THE TOP OF MEMORY
: (NOT TO EXCEED 28K) IS WRITTEN WITH ITS ADDR. THIS
: DATA IS THEN WRITTEN ON THE DISK. THE BUFFER IS
: CLEARED AND THE DATA IS READ BACK AND VERIFIED.
: IN PART TWO, THE EXTENDED ADDRESS BITS ARE TESTED.

```

010526 012767 000004 015142 MEMTST: MOV #4,TESTNO
010534 004567 170524 JSR R5,PRINTS ;PRINT MESSAGE
010540 023740 MES6
010542 016767 015130 171012 MOV TESTNO,TTY
010550 004767 170602 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
010554 012767 177700 015104 MOV #-100,PASSC ;SETUP ITERATION COUNT NO MEM. MANAG.
010562 032767 000500 015042 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
010570 001003 BNE IS ;BR IF TRUE
010572 012767 177750 015066 MOV #-30,PASSC ;IF MEMORY MANAGEMENT DO ONLY 24 PASSES
010600 005067 011146 IS: CLR MEX ;CLEAR DRIVE EXTENDED MEMORY BITS
010604 004767 006032 RMENT: JSR PC,INIT ;INITIALIZE
010610 032767 000100 015014 BIT #86,FLAG ;MEMORY MANAGEMENT?
010616 001520 BEQ RMMEMT ;YES GO DO MEMORY MANAGEMENT TEST
010620 016700 011124 MEMBK: MOV MEMSIZ,R0 ;GET TOP OF CORE
010624 162700 025730 SUB #OUTBUF,R0 ;DETERMINE SIZE OF BUFFER IN BYTES
010630 000241 CLC
010632 006000 ROR R0 ;CONVERT TO WORDS
010634 042700 000001 BIC #1,R0 ;KEEP NUMBER EVEN
010640 010967 014776 MOV R0,WRDCT ;SAVE WORD COUNT OF TRANSFER
010644 012702 025730 7$: MOV #OUTBUF,R2
010650 012703 025730 MOV #OUTBUF,R3 ;GENERATE A PATTERN SO THAT EACH
010654 010322 1$: MOV R3,(R2)+ ;LOCATION CONTAINS ITS ADDRESS
010656 005723 TST (R3)+
010660 020267 011064 CMP R2,MEMSIZ ;HAS ENTIRE PATTERN BEEN GENERATED?
010664 101773 BLOS IS ;BRANCH IF NO
010666 012767 025730 014760 MOV #OUTBUF,BUF ;SET UP BUFFER ADDR
010674 005067 014746 CLR DMA
010700 005067 014740 CLR CYLINDER
010704 004567 004224 JSR R5,FUNCT ;WRITE ADDRESS PATTERN
010710 000003 .WORD 3
010712 105777 014670 TSTB @RPCS ;WAIT FOR DONE
010716 100375 BPL .-4
010720 005777 014662 TST @RPCS ;ANY ERRORS?
010724 100002 BPL 2$ ;BRANCH IF NO
010726 104400 HLT ;ERROR AFTER WRITING ADDR PATTERN
010730 000446 BR 3$
010732 016700 014704 2$: MOV WRDCT,R0
010736 012701 025730 MOV #OUTBUF,R1
010742 005021 10$: CLR (R1)+ ;CLEAR THE BUFFER
010744 005300 DEC R0
010746 001375 BNE 10$
010750 004567 004160 JSR R5,FUNCT ;READ ADDRESS PATTERN
010754 000005 .WORD 5
010756 105777 014624 TSTB @RPCS ;WAIT FOR DONE
010762 100375 BPL .-4

```

```

1464 010764 005777 014616 TST 0RPCS ;ANY ERRORS?
1465 010770 100006 BPL 4$ ;BRANCH IF NO
1466 010772 104400 HLT ;ERROR AFTER READING ADDR PATTERN
1467 010774 032777 040000 014604 BIT #B14,0RPCS ;IS THIS A DATA ERROR?
1468 011002 001401 BEQ 4$ ;BRANCH IF YES
1469 011004 000420 BR 3$
1470 011006 012702 025730 4$: MOV #OUTBUF,R2
1471 011012 010203 MOV R2,R3 ;COMPARE THE ADDR PATTERN
1472 011014 020322 6$: CMP R3,(R2)+ ;IS DATA CORRECT?
1473 011016 001005 BNE 5$ ;BRANCH IF NO
1474 011020 005723 TST (R3)+
1475 011022 020267 010722 CMP R2,MEMSIZ ;IS ENTIRE BUFFER VERIFIED?
1476 011026 101772 BLOS 6$ ;BRANCH IF NO
1477 011030 000406 BR 3$
1478 011032 010367 006720 5$: MOV R3,EXPS
1479 011036 016267 177776 006714 MOV -2(R2),RECS
1480 011044 104401 HLT +1 ;COMPARE ERROR UTILIZING ADDR PATTERN
1481 011046 012767 010644 167726 3$: MOV #7$,LAD ;SETUP LOOP ADD
1482 011054 104000 SCOPE
1483 011056 000527 BR NONEX ;GO DO REST OF TEST
1484
1485 ;THIS PORTION IS FOR MEMORY MANAGEMENT
1486
1487 011060 032767 000400 014544 RMMENT: BIT #B6,FLAG ;BK MACHINE?
1488 011066 001254 BNE MEMBK ;BR IF YES
1489 011070 004767 010670 JSR PC,PARINT ;INITIALIZE KIPARS
1490 011074 004767 010754 1$: TCR PC,PARINC ;INCREMENT KIPARS
1491 011100 004767 011000 JSR PC,PARREG ;GENERATE "MEX" "BUF" "WRDCT"
1492 011104 012700 040000 5$: MOV #40000,R0 ;STARTING ADDRESS OF BUFFER
1493 011110 016701 014526 MOV WRDCT,R1 ;GET BUFFER SIZE
1494 011114 010010 2$: MOV R0,(R0) ;MOVE ADDRESS TO LOCATION
1495 011116 005720 TST (R0)+ ;BUMP LOCATION
1496 011120 005301 DEC R1 ;DONE?
1497 011122 001374 BNE 2$ ;BR IF NO
1498 011124 005067 014516 CLR DMA
1499 011130 005067 014510 CLR CYLINDER
1500 011134 004567 003774 JSR R5,FUNCT ;WRITER ADDRESS PATTERN
1501 011140 000003 .WORD 3
1502 011142 105777 014440 TSTB 0RPCS ;WAIT FOR DONE
1503 011146 100375 BPL -4
1504 011150 005777 014432 TST 0RPCS ;ANY ERRORS?
1505 011154 100005 BPL 3$ ;BR IF NO
1506 011156 104400 HLT ;ERROR WRITTING ADDRESS PATTERN
1507 011160 012767 011104 167614 MOV #5$,LAD ;SETUP LOOP ADDRESS
1508 011166 104000 SCOPE
1509 011170 012700 040000 3$: MOV #40000,R0 ;START OF BUFFER
1510 011174 016701 014442 MOV WRDCT,R1 ;BUFFER SIZE
1511 011200 005020 4$: CLR (R0)+ ;CLEAR BUFFER
1512 011202 005301 DEC R1 ;DONE?
1513 011204 001375 BNE 4$ ;BR IF NO
1514 011206 004567 003722 6$: JSR R5,FUNCT ;READ THE ADDRESS PATTERN
1515 011212 000005 .WORD 5
1516 011214 105777 014366 TSTB 0RPCS ;WAIT FOR DONE
1517 011220 100375 BPL -4
1518 011222 005777 014360 TST 0RPCS ;ANY ERRORS?
1519 011226 !00011 BPL 7$ ;BR IF NO

```

```

1520 011230 104400          HLT                ;DISK ERROR
1521 011232 032777 040000 014346  BIT      #B14,DRPCS ;DATA ERROR?
1522 011240 001404          BEQ      7$         ;BR IF YES
1523 011242 012767 011206 167532  MOV      #6$,LAD    ;SETUP LOOP ADDRESS
1524 011250 104000          SCOPE
1525 011252 012700 040000 7$:  MOV      #40000,R0  ;START OF BUFFER
1526 011256 016701 014360  MOV      WRDCT,R1   ;BUFFER SIZE
1527 011262 010002 9$:  MOV      R0,R2
1528 011264 022002          CMP      (R0)+,R2   ;DATA GOOD?
1529 011266 001004          BNE      8$         ;BR IF DATA NO GOOD
1530 011270 005722          TST      (R2)+
1531 011272 005301          DEC      R1         ;DONE?
1532 011274 001372          BNE      9$         ;BR IF NO
1533 011276 000413          BR       10$        ;CONTINUE THE TST
1534 011300 010267 006452 8$:  MOV      R2,EXPS
1535 011304 014067 006450  MOV      -(R0),RECS
1536 011310 104405          HLT      +5         ;DATA COMPARE ERROR
1537 011312 012767 011262 167462  MOV      #9$,LAD    ;SETUP LOOP ADDRESS
1538 011320 104000          SCOPE
1539 011322 005720          TST      (R0)+     ;TEST REST OF BUFFER
1540 011324 000756          BR       9$         ;TEST MORE
1541 011326 032767 001000 014276 10$: BIT      #B9,FLAG  ;LAST PAGE?
1542 011334 001657          BEQ      1$         ;BR IF NO
1543          ;CHECK THE SETTING OF NON EXISTENT MEMORY FLAG IN RPER.
1544
1545 011336 032767 000100 014266 NONEX: BIT      #B6,FLAG ;IS MEMORY MANAGEMENT ON?
1546 011344 001002          BNE      4$         ;BR IF NOT ON
1547 011346 005037 177572          CLR      #SRO      ;IF ON TURN OFF
1548 011352 012767 000001 014262 4$:  MOV      #1,WRDCT  ;SETUP WORD COUNT
1549 011360 012767 160000 014266  MOV      #160000,BUF ;LOAD ADDR TO FORCE NON EX MEMORY
1550 011366 004567 003542          JSR      RS,FUNCT  ;ISSUE A WRITE COMMAND AND
1551 011372 000063          .WORD 63          ;EXPECT NON EX MEMORY
1552 011374 105777 014206          TSTB    DRPCS      ;WAIT FOR READY
1553 011400 100375          BPL      -4
1554 011402 032777 000004 014214  BIT      #B2,DRPER ;DID NON EX MEMORY SET
1555 011410 001002          BNE      1$         ;BRANCH IF SET
1556 011412 104400          HLT
1557 011414 000413          BR       2$         ;NON EX MEMORY DID NOT SET
1558 011416 032777 040000 014162 1$:  BIT      #B14,DRPCS ;DID HARD ERROR SET ON NON EX MEMORY
1559 011424 001002          BNE      3$         ;BRANCH IF SET
1560 011426 104400          HLT                ;HARD ERROR DID NOT SET
1561 011430 000405          BR       2$
1562 011432 032777 100000 014146 3$:  BIT      #B15,DRPCS ;DID THE ERROR FLAG SET AFTER NON EX MEMORY
1563 011440 001001          BNE      2$         ;BRANCH IF SET
1564 011442 104400          HLT                ;ERROR FLAG DID NOT SET
1565 011444 012767 011352 167330 2$:  MOV      #4$,LAD
1566 011452 104000          SCOPE
1567
1568          ;IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS
1569          ;BITS.
1570
1571 011454 012767 012252 166322 EXTTST: MOV      #EXTTRP,4 ;SETUP TIMEOUT TRAP
1572 011462 012767 000340 166316  MOV      #PRI7,6
1573 011470 005737 177572          TST      #SRO
1574
1575          ;IF MEMORY MANAGEMENT IS NOT
          ;AVAILABLE THE PROGRAM WILL TRAP
          ;AND TRANSFER TO END OF THE TEST

```


1576	011474	012767	012236	166302		MOV	#EXTRP,4	
1577	011502	012737	007600	172356		MOV	#7600, @#KIPAR7	; OPEN I/O REGISTERS
1578	011510	005037	172340			CLR	@#KIPAR0	; FREE FIRST 4K
1579	011514	012737	000200	172342		MOV	#200, @#KIPAR1	; ENABLE SECOND 4K
1580	011522	012737	002000	172344		MOV	#2000, @#KIPAR2	
1581	011530	012737	177406	172300		MOV	#400*256.-400+UP+RW, @#KIPDR0	; SET KIPDR0=RW UP 400 BLOCKS
1582	011536	012737	177406	172302		MOV	#400*256.-400+UP+RW, @#KIPDR1	; SET KIPDR1=RW UP 400 BLOCKS
1583	011544	012737	177406	172304		MOV	#400*256.-400+UP+RW, @#KIPDR2	; SET KIPDR2=RW UP 400 BLOCKS
1584	011552	012737	177406	172316		MOV	#400*256.-400+UP+RW, @#KIPDR7	; SET KIPDR7=RW UP 400 BLOCKS
1585	011560	012737	000001	177572		MOV	#1, @#SRO	; TURN ON MEMORY MANAGEMENT
1586	011566	012702	040000			MOV	#40000, R2	; R2 EQUALS BASE ADDR
1587	011572	012712	177777		7\$:	MOV	#177777, (R2)	; INSERT PATTERN INTO 200000
1588	011576	012767	000002	014036		MOV	#2, WRDCT	; SETUP WORDCOUNT
1589	011604	005067	010142			CLR	MEX	; CLEAR EXTENDED MEMORY BITS FOR CONTROLER
1590	011610	012767	177777	014036		MOV	#177777, BUF	; SETUP BUS ADDR
1591	011616	004567	003312			JSR	RS, FUNCT	; WRITE TWO WORDS ON DISK. RPBA
1592	011622	000003			.WORD	3		; STARTS AT 177777 TO FORCE CARRY
1593								; TO SET MEX0
1594	011624	105777	013756			TSTB	@RPCS	; WAIT FOR READY
1595	011630	109375				BPL	.-4	
1596	011632	005777	013750			TST	@RPCS	
1597	011636	100002				BPL	1\$	
1598	011640	104400				HLT		; STATUS ERROR AFTER 2 WORD WRITE
1599	011642	000447				BR	2\$; USING MEX0
1600	011644	032777	000020	013734	1\$:	BIT	#B4, @RPCS	; MEX0 SHOULD HAVE SET?
1601	011652	001002				BNE	3\$; BRANCH IF SET
1602	011654	104400				HLT		; MEX0 DID NOT SET
1603	011656	000441				BR	2\$	
1604	011660	005012			3\$:	CLR	(R2)	; CLEAR LOCATION 200000
1605	011662	004567	003246			JSR	RS, FUNCT	; READ TWO WORDS INTO LOCATIONS
1606	011666	000005			.WORD	5		; 177777 AND 200000.
1607	011670	105777	013712			TSTB	@RPCS	; WAIT FOR READY
1608	011674	100375				BPL	.-4	
1609	011676	005777	013704			TST	@RPCS	; ANY ERRORS?
1610	011702	100002				BPL	4\$; BRANCH IF NO
1611	011704	104400				HLT		; ERROR OFTER READING 2 WORDS
1612	011706	000425				BR	2\$	
1613	011710	032777	000020	013670	4\$:	BIT	#B4, @RPCS	; DID MEX0 SET?
1614	011716	001002				BNE	5\$; BRANCH IF YES
1615	011720	104400				HLT		; MEX0 DID NOT SET AFTER 2 WORD
1616	011722	000417				BR	2\$; READ STARTING AT 177777
1617	011724	022712	177777		5\$:	CMP	#177777, (R2)	; WAS DATA READ INTO LOCATION
1618	011730	001407				BEQ	6\$; 200000 CORRECTLY? - BRANCH IF YES
1619	011732	012767	177777	006016		MOV	#177777, EXPS	
1620	011740	011267	006014			MOV	(R2), RECS	
1621	011744	104401				HLT	+1	; DATA COMPARE ERROR AT 200000
1622	011746	000405				BR	2\$; IF RECEIVED=0 - LOCATION WASN'T ACCESSED
1623	011750	032777	000040	013630	6\$:	BIT	#B5, @RPCS	; MEX1 SHOULD BE CLEAR
1624	011756	001401				BEQ	2\$; BRANCH IF CLEAR
1625	011760	104400				HLT		; MEX1 IS SET - SHOULD NOT BE
1626	011762	012767	011572	167012	2\$:	MOV	#7\$, LAD	; SETUP ERROR LOOP
1627	011770	104000				SCOPE		
1628	011772	012737	004000	172344	EXTT1:	MOV	#4000, @#KIPAR2	
1629	012000	012702	040000			MOV	#40000, R2	; R2 EQUALS THE BASE ADDR
1630	012004	012712	177777		7\$:	MOV	#177777, (R2)	; INSERT PATTERN INTO 400000
1631	012010	012767	177777	013636		MOV	#177777, BUF	; SETUP BUS ADDR

1632	012016	004567	003112		JSR	R5,FUNCT	
1633	012022	000023		.WORD	23		
1634	012024	105777	013556		TSTB	2RPCS	;WAIT FOR READY
1635	012030	100375			BPL	-4	
1636	012032	005777	013550		TST	2RPCS	;ANY ERRORS?
1637	012036	100002			BPL	45	;BRANCH IF NO
1638	012040	104400			HLT		
1639	012042	000455			BR	25	;ERROR AFTER READING 2 WORDS
1640	012044	032777	000020	013534	45:	BIT	#84,2RPCS
1641	012052	001402			BEQ	55	;DID MEXO CLEAR?
1642	012054	104400			HLT		;BRANCH IF YES
1643	012056	000447			BR	25	;MEXO DID NOT CLEAR AFTER 2 WORD
1644	012060	032777	000040	013520	55:	BIT	#85,2RPCS
1645	012066	001002			BNE	105	;READ STARTING AT 377777
1646	012070	104400			HLT		;DID MEXI SET?
1647							;BRANCH IF YES
1648	012072	000441			BR	25	;MEXI DI NOT SET WITH A TWO WORD
1649	012074	005012			CLR	(R2)	;TRANSFER STARTING AT 377777
1650	012076	004567	003032		JSR	R5,FUNCT	;CLEAR LOCATION 400000
1651	012102	000025		.WORD	25		;READ TWO WORDS STARTING AT 377777
1652	012104	105777	013476		TSTB	2RPCS	
1653	012110	100375			BPL	-4	;WAIT FOR READY
1654	012112	005777	013470		TST	2RPCS	
1655	012116	100002			BPL	115	;ANY ERRORS?
1656	012120	104400			HLT		;BRANCH IF NO
1657	012122	000425			BR	25	;ERROR WHILE READING TWO WORDS
1658	012124	032777	000020	013454	115:	BIT	#84,2RPCS
1659	012132	001402			BEQ	125	;DID MEXO CLEAR?
1660	012134	104400			HLT		;BRANCH IF YES
1661	012136	000417			BR	25	;MEXO DID NOT CLEAR AFTER 2 WORD
1662	012140	022712	177777		125:	CMP	#177777,(R2)
1663	012144	001407			BEQ	65	;READ STARTING AT 377777
1664	012146	012767	177777	005602		MOV	#177777,EXPS
1665	012154	011267	005600			MOV	(R2),RECS
1666	012160	104401			HLT	+1	;DATA COMPARE ERROR AT 400000 IF
1667	012162	000405			BR	25	;RECEIVED=0 - LOCATION WASN'T ACCESSED
1668	012164	032777	000040	013414	65:	BIT	#85,2RPCS
1669	012172	001001			BNE	25	;DID MEXI SET?
1670	012174	104400			HLT		;MEXI DID NOT SET AFTER 2 WORD TRANSFER
1671							;STARTING AT 377777
1672	012176	012767	012004	166576	25:	MOV	#75,LAD
1673	012204	104000			SCOPE		;SETUP ERROR LOOP
1674	012206	000413			BR	EXTRP	;CLEAR MEMORY MANAGEMENT
1675	012210	005267	013452		EXTEND:	INC	;INCREMENT ITERATION COUNT
1676	012214	001402			BEQ	15	
1677	012216	000167	176362		JMP	RMENT	
1678	012222	032737	002000	177570	15:	BIT	#B10,2#SWR
1679	012230	001420			BEQ	DATA1	;LOOP ON TEST?
1680	012232	000167	176346		JMP	RMENT	
1681							
1682	012236	032767	000100	013366	EXTRP:	BIT	#86,FLAG
1683	012244	001402			BEQ	EXTRP	;USING MEMORY MANAGEMENT?
1684	012246	005037	177572		CLR	2#SR0	;BR IF YES
1685	012252	012706	000500		EXTTRP:	MOV	;NO TURN IT OFF
1686	012256	012767	000006	165520		MOV	#STKPTR,SP
1687	012264	005067	165516		CLR	#6,4	;RESTORE STACK

RP11C RELIABILITY TEST MACY11 27(732) 04-NOV-76 14:18 PAGE 35
DZRPB.P11 ***** TEST 4 *****

1698 012270 000747

BN EXTEND

```

1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702 012272 012767 000005 013376
1703 012300 004567 166760
1704 012304 023740
1705 012306 016767 013364 167246
1706 012314 004767 167036
1707 012320 005067 013376
1708 012324 005067 007422
1709 012330 016700 013316
1710 012334 000241
1711 012336 006000
1712 012340 010037 177570
1713 012344 005067 013274
1714 012350 005067 013272
1715 012354 012737 000200 177776
1716 012362 016767 013270 013252
1717 012370 012777 015224 013204
1718 012376 012777 000340 013200
1719 012404 032767 000500 013220
1720 012412 001006
1721 012414 004767 007344
1722 012420 000240
1723 012422 000240
1724 012424 004767 007454
1725 012430 004767 004404
1726 012434 032767 000500 013170
1727 012442 001403
1728 012444 012767 025730 013202
1729 012452 032767 040000 013152
1730 012460 001426
1731 012462 004767 003434
1732 012466 004767 003126
1733 012472 004567 002436
1734 012476 000103
1735 012500 032737 010000 177570
1736 012506 001003
1737 012510 004767 007534
1738 012514 000401
1739 012516 000001
1740 012520 012767 012462 166254
1741 012526 104000
1742 012530 004767 003424
1743 012534 000752
1744 012536 032767 020000 013066

```

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

```

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

```

```

DATAT: MOV #5,TESTNO
      JSR RS,PRINTS ;PRINT MESSAGE
      MES6
      MOV TESTNO,TTY
      JSR PC,PRINTS ;TYPE LOCATION-SUPPRESS ZEROS
      CLR CNTA ;INITIALIZE READ COUNTER
      CLR MEX ;CLEAR EXTENDED MEMORY BITS IN CONTROLER
RDATAT: MOV PATNU,RO ;GET PATTERN NO.
      CLC
      ROR RO
      MOV RO,@#SWR ;DISPLAY PATTERN NO. IN USE
      CLR CYLINDER
      CLR DMA
      MOV #PRI4,@#PSW ;ENABLE INT SYSTEM
      MOV SWRDCT,WRDCT
      MOV #DKINT,@VECTOR ;SETUP DISK VECOTR
      MOV #340,@STATUS
      BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG
      BNE DATP ;BR IF TRUE
      JSR PC,PARINT ;INITIALIZE KIPARS
      NOP
      NOP ;NEEDED FOR CONFORMITY
      JSR PC,PARREG ;SETUP "MEX","WRDCT","BUF"
DATP: JSR PC,PASEL ;GENERATE PATTERN
      BIT #500,FLAG ;BK MEM. MANAG. OR NO MEM. MANAG?
      BEQ IS ;BR IF NOT TRUE
      MOV #OUTBUF,BUF ;SETUP BUFFER ADDR
      BIT #B14,FLAG ;WRITE?
      BEQ WRICK ;BRANCH IF NO
LDAT: JSR PC,OPDSEL ;ANY OPERATOR ADDR PARAMETERS?
      JSR PC,SEEK ;GO DO A RANDOM SEEK
      JSR RS,FUNCT ;WRITE WITH INTERRUPTS
      .WORD 103
      BIT #B12,@#SWR ;DETERMINE HOW TO WAIT FOR INT
      BNE IS
      JSR PC,NPR ;GENERATE WORSE CASE NPR CYCLES
      BR 2S
      1S: WAIT
      2S: MOV #LDAT,LAD ;SETUP LOOP ADDR
      SCOPE
      JSR PC,DISBUF ;PREPARE NEW DISK ADDR
      BR LDAT
WRICK: BIT #B13,FLAG ;WRITE CHECK?

```


1801	013024	005021			CLR	(R1)+	
1802	013026	005021			CLR	(R1)+	
1803	013030	005021			CLR	(R1)+	
1804	013032	005021			CLR	(R1)+	
1805	013034	005021			CLR	(R1)+	
1806	013036	160200			SUB	R2,RO	;DECREMENT WORD COUNT
1807	013040	001354			SNE	4\$	
1808	013042	004767	004750	10\$:	JSR	PC,COMPAR	;COMPARE THE DATA
1809	013046	105777	012534		TSTB	DRPCS	;WAIT FOR READY
1810	013052	100375			BPL	-4	
1811	013054	005767	166160	3\$:	TST	ERRFLG	;WERE THERE ANY ERRORS
1812	013060	001424			BEQ	5\$;BRANCH IF NO
1813	013062	005267	012604		INC	RDERR	;UPDATE ERROR COUNT
1814	013066	022767	000024	012576	CMP	#20.,RDERR	;MORE THAN 20 ERRORS?
1815	013074	001416			BEQ	5\$;BRANCH IF YES
1816	013076	022767	000012	012566	CMP	#10.,RDERR	;IS THIS TENTH ERROR?
1817	013104	001274			BNE	DSKRD	;BRANCH IF NO
1818	013106	112777	000015	012472	MOVB	#15,DRPCS	;HOME THE HEADS
1819	013114	105777	012466		TSTB	DRPCS	;WAIT FOR DONE
1820	013120	100375			BPL	-4	
1821	013122	005777	012500		TST	DRPDS	
1822	013126	100375			BPL	-4	;WAIT FOR READY
1823	013130	000662			BR	DSKRD	
1824	013132	005767	012534	5\$:	TST	RDERR	
1825	013136	001417			BEQ	6\$	
1826	013140	022767	000013	012524	CMP	#11.,RDERR	;WAS READ ERROR STILL THERE AFTER A RECALL
1827	013146	002003			BGE	9\$;BR IF NO
1828	013150	012767	000002	012544	MOV	#2,CNTA	;YES DON'T TRY TO READ AGAIN UNLESS LOOPING
1829	013156			9\$:			
1830	013156	004567	166102		JSR	R5,PRINT\$;PRINT MESSAGE
1831	013162	023753			MES7		
1832	013164	016767	012502	166370	MOV	RDERR,TTY	
1833	013172	004767	166160		JSR	PC,PRINTS	;TYPE LOCATION-SUPRESS ZEROS
1834	013176	005067	012470	6\$:	CLR	RDERR	;CLEAR READ ERROR COUNTER
1835	013202	012767	012662	165572	MOV	#ESH,LAD	;LOOP ADDR
1836	013210	104000			SCOPE		
1837	013212	005267	012504		INC	CNTA	;INCREMENT READ COUNTER
1838	013216	022767	000003	012476	CMP	#3,CNTA	;DONE 3 READS?
1839	013224	001216			BNE	ESH	;BR IF NO
1840	013226	032767	000500	012376	BIT	#500,FLAG	;BK MEM. MANAG. OR NO MEM. MANAG.
1841	013234	001012			BNE	8\$;BR IF TRUE
1842	013236	032767	001000	012366	BIT	#89,FLAG	;LAST PAGE TRANSFERED?
1843	013244	001402			BEQ	7\$;BR IF NO
1844	013246	004767	006512		JSR	PC,PARINT	;GO REINTIALIZE KIPARS
1845	013252	004767	006576	7\$:	JSR	PC,PARINC	;GO INCREMENT KIPARS BY 1K
1846	013256	004767	006622		JSR	PC,PARREG	;GO SETUP "MEX" "WRDCT", "BUF"
1847	013262	004767	002672	8\$:	JSR	PC,DISBUF	;GET NEW DISK ADDR
1848	013266	000401			BR	11\$	
1849	013270	000402			BR	MSTR	
1850	013272	000167	177360	11\$:	JMP	READ1	
1851	013276	032767	000040	012326	BIT	#85,FLAG	;LOOPING ON AN OPERATOR ADDR?
1852	013304	001402			BEQ	1\$;NO CONTINUE
1853	013306	000167	177116		JMP	DATP	;YES
1854	013312	005767	012314	1\$:	TST	FLAG	
1855	013316	100002			BPL	2\$;UNDER PROGRAM CONTROL
1856	013320	000167	001034		JMP	MULCHK	;OPERATOR SELECTED PATTERN

1857	013324	062767	000002	012320	2\$:	ADD	#2,PATNU	;INC PATTERN INDEX
1858	013332	022767	000036	012312		CMP	#36,PATNU	;PATTERNS EXCEEDED?
1859	013340	001402				BEQ	3\$	
1860	013342	000167	176762			JMP	RDATAT	;NOT YET
1861	013246	005067	012300		3\$:	CLR	PATNU	;LAST PATTERN USED
1862	013352	032737	002000	177570		BIT	#810,2\$SWR	;LOOP ON TEST?
1863	013360	001402				SEQ	RANEX	;NO..GO TO RANDOM TEST
1864	013362	000167	176742			JMP	RDATAT	;YES

```

1865
1866
1867
1868
1869
1870
1871 013366 012767 000006 012302 RANEX: MOV #6,TESTNO
1872 013374 004567 165664 JSR RS,PRINTS ;PRINT MESSAGE
1873 013400 023740 MES6
1874 013402 016767 012270 166152 MOV TESTNO,TTY
1875 013410 004767 165742 JSR PC,PRINTS ;TYPE LOCATION-SUPRESS ZEROS
1876 013414 005067 006332 CLR MEX ;CLEAR EXTENDED MEMORY BITS IN CONTROLER
1877 013420 004767 003216 RRANEX: JSR PC,INIT
1878 013424 032767 000503 012200 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.
1879 013432 001004 BNE 1$ ;BR IF TRUE
1880 013434 004767 006324 JSR PC,PARINT ;INITALIZE ALL KIPARS
1881 013440 000240 NOP ;NEEDED FOR CONFORMITY
1882 013442 000240 NOP
1883 013444 012767 000034 012200 1$: MOV #34,PATNU
1884 013452 012767 175300 012206 MOV #-2500,PASSC ;SET UP PASS COUNT
1885 013460 012737 000200 177776 MOV #PRI4,#PSW
1886 013466 012767 000700 012146 WRLG: MOV #700,WRDCT ;SET UP WORD COUNT TO 1+ SECTOR
1887 013474 016767 012142 012200 MOV WRDCT,WORK
1888 013502 012701 025730 MOV #OUTBUF,R1
1889 013506 022767 000500 012116 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
1890 013514 001002 BNE 4$ ;BR IF TRUE
1891 013516 012701 040000 MOV #40000,R1 ;START OF OUTBUFF WITH MEMORY MANAG.
1892 013522 004767 003420 4$: JSR PC,RANDOM ;GENERATE RANDOM PATTERN
1893 013526 1$:
1894 013526 004767 166032 JSR PC,RAND$ ;GENERATE TWO RANDOM NOS.
1895 013532 016767 166144 012142 MOV LONUM,WORK
1896 013540 016767 166134 012136 MOV HINUM,WORK1
1897 013546 042767 177000 012126 BIC #177000,WORK
1898 013554 022767 000625 012120 CMP #625,WORK ;FORM RANDOM CYL ADDR
1899 013562 002761 1$:
1900 013564 016767 012112 012052 MOV WORK,CYLINDER ;SAVE IT
1901 013572 042767 160360 012104 BIC #160360,WORK1
1902 013600 122767 000010 012076 CMPB #10,WORK1 ;FORM RANDOM SECTOR ADDR
1903 013606 101003 BHI 2$
1904 013610 042767 000010 012066 BIC #10,WORK1
1905 013616 122767 000023 012061 2$: CMPB #23,WORK1+1
1906 013624 101003 BHI 3$ ;FORM RANDOM HEAD ADDR
1907 013626 142767 000014 012051 BICB #14,WORK1+1
1908 013634 016767 012044 012004 3$: MOV WORK1,DMA ;SAVE DESK ADDR.
1909 013642 012767 025730 012004 RANLOP: MOV #OUTBUF,BUF ;SETUP OUTPUT BUFFER
1910 013650 032767 000500 011754 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.?
1911 013656 001003 BNE 13$ ;BR IF TRUE
1912 013660 012767 040000 011766 MOV #40000,BUF ;START OF OUTBUFFER WITH MEMORY MANAG.
1913 013666 004767 001726 13$: JSR PC,SEEK ;DO A RANDOM SEEK
1914 013672 004567 001236 JSR RS,FUNCT ;WRITE RANDOM DATA AND
1915 013676 000103 .WORD 103 ;ENABLE INTERRUPTS
1916 013700 032737 010000 177570 BIT #B12,#SWR ;DETERMINE HOW TO WAIT FOR INT
1917 013706 001003 BNE 2$
1918 013710 004767 006334 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
1919 013714 000401 BR 4$
1920 013716 000001 2$: WAIT

```



```

1921 013720 012767 013642 165054 48: MOV #RANL0P,LAD ;SETUP LOOP ADDR
1922 013726 104000 SCOPE
1923 013730 004567 001200 78: JSR R5,FUNCT ;WRITE CHECK THE DATA AND
1924 013734 000107 .WORD 107 ;ENABLE INTERRUPT
1925 013736 032737 010000 177570 BIT #B12,2#SWR ;HOW TO WAIT FOR INT?
1926 013744 001003 BNE 15 ;
1927 013746 004767 006276 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
1928 013750 000401 BR 55 ;
1929 013754 000001 18: WAIT
1930 013756 012767 013730 165016 58: MOV #78,LAD ;SETUP LOOP ADDR
1931 013764 104000 SCOPE
1932 013766 005067 011730 CLR CNTA ;INITALIZE READ COUNTER
1933 013772 005067 011674 68: CLR RDERR ;CLEAR READ ERROR COUNTER
1934 013776 004767 001616 JSR PC,SEEK ;DO A RANDOM SEEK
1935 014002 005067 170472 118: CLR INTFLG
1936 014006 004567 001122 JSR R5,FUNCT ;READ RANDOM DATA AND
1937 014012 000105 .WORD 105 ;ENABLE INTERRUPT
1938 014014 012700 000700 MOV #700,R0 ;SET UP TO CLEAR INPUT BUFFER
1939 014020 012702 030020 MOV #16,R2
1940 014024 012701 025730 MOV #OUTBUF,R1 ;START OF INPUT BUFFER
1941 014030 032767 000500 011574 BIT #500,FLAG ;BK WITH MEM. MANAG. OR NO MEM MANAG?
1942 014036 001002 BNE 15 ;BR IF TRUE
1943 014040 012701 040000 MOV #40000,R1 ;START OF OUTBUF WITH MEM. MANAG.
1944 014044 158:
1945 014044 005021 CLR (R1)+
1946 014046 005021 CLR (R1)+
1947 014050 005021 CLR (R1)+
1948 014052 005021 CLR (R1)+
1949 014054 005021 CLR (R1)+
1950 014056 005021 CLR (R1)+
1951 014060 005021 CLR (R1)+
1952 014062 005021 CLR (R1)+
1953 014064 005021 CLR (R1)+
1954 014066 005021 CLR (R1)+
1955 014070 005021 CLR (R1)+
1956 014072 005021 CLR (R1)+
1957 014074 005021 CLR (R1)+
1958 014076 005021 CLR (R1)+
1959 014100 005021 CLR (R1)+
1960 014102 005021 CLR (R1)+
1961 014104 160200 SUB R2,R0 ;DECREMENT THE WORD COUNT
1962 014106 001356 BNE 15 ;GO CLR MORE
1963 014110 032737 010000 177570 BIT #B12,2#SWR ;HOW TO WAIT FOR INT?
1964 014116 001003 BNE 35 ;
1965 014120 004767 006124 JSR PC,NPR ;TEST WORSE CASE NPR CYCLES
1966 014124 000401 BR 65 ;
1967 014126 000001 38: WAIT
1968 014130 032737 001000 177570 68: BIT #B9,2#SWR ;COMPARE FOR ERRORS?
1969 014136 001006 BNE 95 ;BRANCH IF NO
1970 014140 032777 040000 011440 BIT #B14,2#RPCS ;HARD ERROR?
1971 014146 001002 BNE 95 ;BRANCH IF YES
1972 014150 004767 003642 JSR PC,COMPARE ;COMPARE DATA FOR ERRORS
1973 014154 005767 165060 98: TST ERRFLG ;READ ERROR?
1974 014160 001424 BEQ 105 ;BRANCH IF NO
1975 014162 005267 011504 INC RDERR ;UPDATE ERROR COUNT
1976 014166 022767 000024 011476 CMP #20,.RDERR ;20 ERRORS YET?
  
```

1977	014174	001416			BEQ	10\$: BRANCH IF YES
1978	014176	022767	000012	011466	CMP	#10.,RDERR		: IS THIS TENTH ERROR?
1979	014204	001276			BNE	11\$: BRANCH IF NO
1980	014206	112777	000015	011372	MOVB	#15,DRPCS		: ISSUE HOME COMMAND
1981	014214	105777	011366		TST	DRPCS		: WAIT FOR DONE
1982	014220	100375			BPL	-4		
1983	014222	005777	011400		TST	DRPDS		: WAIT FOR READY
1984	014226	100375			BPL	-4		
1985	014230	000554			BR	11\$		
1986	014232	005767	011434		TST	RDERR		
1987	014236	001417			BEQ	21\$		
1988	014240	022767	000013	011424	CMP	#11.,RDERR		: DID RECALL HELP RECOVERY
1989	014246	002003			BGE	14\$: BR IF YES
1990	014250	012767	000011	011444	MOV	#9.,CNTA		: NO, DON'T TRY READING AGAIN
1991	014256							
1992	014256	004567	165002		JSR	R5,PRINTS		: PRINT MESSAGE
1993	014262	023753			MES7			
1994	014264	016767	011402	165270	MOV	RDERR,TTY		
1995	014272	004767	165060		JSR	PC,PRINTS		: TYPE LOCATION-SUPRESS ZEROS
1996	014276	005067	011370		CLR	RDERR		: CLEAR READ ERROR COUNTER
1997	014302	012767	013772	164472	MOV	#8\$,LAD		: SET UP LOOP ADDR
1998	014310	104000			SCOPE			
1999	014312	005267	011404		INC	CNTA		: INCREMENT READ COUNTER
2000	014316	022767	000012	011376	CMP	#10.,CNTA		: DONE 10 DISK READS?
2001	014324	001222			BNE	8\$: BR IF NO
2002	014326	005267	011334		INC	PASSC		: INCREMENT PASS COUNT
2003	014332	001402			BEQ	12\$: BRANCH IF DONE
2004	014334	000167	177126		JMP	WALG		: CONTINUE
2005	014340	005067	011306		CLR	PATNU		
2006	014344	032737	002000	177570	BIT	#810,DRSWR		: LOOP ON TEST?
2007	014352	001402			BEQ	MULCHK		: NO
2008	014354	000167	177040		JMP	RRANEX		: LOOP
2009								
2010								: CHECK FOR MULTI DISK MODE
2011								: IF IN MULTI DISK MODE REPORT "END"
2012								: IF LAST DISK ON SYSTEM HAS BEEN
2013								: EXERCISED.
2014								
2015								
2016	014360	005067	011262		MULCHK: CLR	DMA		
2017	014364	005067	011254		CLR	CYLINDER		: CLEAR ADDRESS REGISTERS
2018	014370	032767	004000	011234	BIT	#811,FLAG		: ARE WE IN MULTI DISK MODE
2019	014376	001422			BEQ	REPOEN		: REPORT "END"
2020	014400	016767	011226	011274	MOV	FLAG,WORK		: WHAT DISK ARE WE ON
2021	014406	042767	177743	011266	BIC	#177743,WORK		: IF LAST DISK ON SYSTEM
2022	014414	026767	011262	011240	CMP	WORK,DSKNOR		: REPORT END
2023	014422	001004			BNE	INDRVE		
2024	014424	042767	000034	011200	BIC	#34,FLAG		
2025	014432	000404			BR	REPOEN		: REPORT "END" LAST DISK
2026	014434	062767	000004	011170	INDRVE: ADD	#4,FLAG		: INC. DISK NO.
2027	014442	000422			BR	EXTPP		: EXERCISE DISK
2028	014444	005267	164330		REPOEN: INC	ICNT		: INCREMENT PASS COUNTER
2029	014450	004567	164610		JSR	R5,PRINTS		: PRINT MESSAGE
2030	014454	023550			MES1			: REPORT END OF PASS
2031	014456	016767	164316	165076	MOV	ICNT,TTY		
2032	014464	004767	164666		JSR	PC,PRINTS		: TYPE LOCATION-SUPRESS ZEROS

2033	014470	013701	000042		MUV	0#42,R1	;GET MONITOR RETURN ADDRESS
2034	014474	001405			BEO	EXTPP	;BRANCH IF NOT UNDER MONITOR
2035	014476	000005			RESET		
2036	014500	004711		MEXIT:	JSR	PC,(R1)	;EXIT TO THE MONITOR
2037	014502	000240			NOP		
2038	014504	000240			NOP		
2039	014506	000240			NOP		
2040	014510	000167	167032	EXTPP:	JMP	ADTST	;RECYCLE
2041							
2042							

.SBTTL ***** TEST 7 *****

:TEST THE ABILITY OF THE RP11C TO SENSE POWER FAILURE
:AND TO HOME THE HEADS. WHEN POWER IS RESTORED
:THE CYLINDER ADDRESS IS TESTED FOR ZERO. AFTER TYPING THE MESSAGE
:REQUESTING POWER TO BE TURNED OFF THE PROGRAM GOES INTO
:A LOOP READING FROM THE DISK. AFTER POWER IS RESTORED,
:MEMORY IS CHECKED TO SEE THAT THE DISK DID NOT PUT ANY
:JUNK INTO MEMORY WHILE POWER WAS GOING DOWN.

2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098

014514 012706 000500
014520 012767 015104 163276
014526 012767 000340 163272
014534 004567 002030
014540 012777 000625 011050
014546 052777 000011 011032
014554 012700 025730
014560 012720 025252
014564 020067 005160
014570 101773
014572 012767 000625 011044
014600 005067 011042
014604 012767 000400 011030
014612 012767 025730 011034
014620 105777 010762
014624 100375
014626 005777 010774
014632 100375
014634 004567 000274
014640 000003
014642 105777 010740
014646 100375
014650 032777 100000 010730
014656 001401
014660 104400
014662 012767 014514 164112
014670 104000
014672 004567 164366
014676 023672
014700 004567 000230
014704 000005
014706 105777 010674
014712 100375
014714 000771

014716 012777 000001 010662
014724 016767 010702 010750
014732 000241
014734 006067 010742
014740 006067 010736
014744 000367 010732
014750 042767 174377 010724

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

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

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

2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193

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

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

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

.SBTTL *** SUBROUTINES ***

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

```
FUNCT: JSR    R5, DSKNOS      ; SELECT THE UNIT
        MOV    DMA, @RPDA     ; SETUP DISK ADDR REG
        MOV    CYLINDER, @RPGA ; SETUP CYLINDER ADDR REG
        MOV    BUF, @RPBA     ; SETUP BUS ADDR REG
        MOV    WRDCT, @RPWC   ; SETUP WORD COUNT
        NEG    @RPWC         ; COMPLIMENT WORD COUNT
        MOV    (R5), FNCT     ; GET RPCS FUNCTION
        ADD    #2, R5        ; UPDATE RETURN ADDR
        BIS    MEX, FNCT     ; ADD EXTENDED MEMORY BITS
        BIS    FNCT, @RPCS   ; OUTPUT THE FUNCTION
        RTS    R5
FNCT: 0
```

; RP11 DISK INTERRUPT HANDLER

```
DKINT: CLR    ERRFLG        ; CLEAR THE ERROR FLAG
        TST    @RPCS       ; TEST FOR ERROR
        BMI    1$
        JMP    INTXT       ; JUMP IF NO ERRORS
1$:    MOV    SP, ERRFLG   ; SET INTERRUPT ERROR FLAG
        TST    RDERR      ; IS THIS THE FIRST ERROR ATTEMPT?
        BEQ    2$        ; BRANCH IF YES
        BIT    #84, @SWR  ; TYPE ALL ERROR ATTEMPTS?
        BEQ    DK11      ; BRANCH IF NO
2$:    HLT
        BIT    #81, @RPCS ; CHECK FOR READ
        BNE    DELMES    ; BRANCH IF WRITTING
        JSR    R5, PRINT$ ; PRINT MESSAGE
        MOV    MES13
        JSR    RDERR, TTY ; GIVE # OF READ ATTEMPT
        JSR    PC, PRINTS ; TYPE LOCATION-SUPRESS ZEROS
        MOV    SP, ERRFLG
DELMES: MOV    @RPGA, INT1 ; GET CYLINDER ADDR
        MOV    @RPDA, INTO ; GET HEAD AND SECTOR ADDR
        BIC    #160360, INTO ; CLEAR UNWANTED BITS
        BIT    #80, @RPER  ; WAS IT AN ADDR ERROR?
        BNE    REDAC     ; BRANCH IF YES
        BIT    #17, INTO  ; IS SECTOR = TO 0
        BEQ    DECTK     ; YES - BRANCH
        DEC    INTO      ; BACK UP COUNT
        BR    REDAC
DECTK: BITB  #37, INTO+1 ; IS HEAD = TO 0
        BEQ    DECCY    ; YES - BRANCH
        DECB INTO+1    ; BACK UP HEAD
        BIS    #11, INTO ; SET UP SECTOR
        BR    REDAC
DECCY: MOV    #11411, INTO
        DEC    INT1
REDAC:
```

```

2194 015430 004567 163630          JSR    RS,PRINT$      ;PRINT MESSAGE
2195 015434 024115          MES14                ;REPORT CYLINDER ADDR
2196 015436 016767 000154 164116  MOV    INT1,TTY
2197 015444 004767 163706          JSR    PC,PRINT$     ;TYPE LOCATION-SUPRESS ZEROS
2198 015450 005067 000142          CLR    INT1
2199 015454 116767 000135 000134  MOVB   INTO+1,INT1
2200 015462 004567 163576          JSR    RS,PRINT$     ;PRINT MESSAGE
2201 015466 024125          MES15                ;REPORT HEAD ADDR OF FAILURE
2202 015470 016767 000122 164064  MOV    INT1,TTY
2203 015476 004767 163654          JSR    PC,PRINT$     ;TYPE LOCATION-SUPRESS ZEROS
2204 015502 116767 000106 000106  MOVB   INTO,INT1
2205 015510 004567 163550          JSR    RS,PRINT$     ;PRINT MESSAGE
2206 015514 024136          MES16                ;REPORT SEC ADDR OF FAILURE
2207 015516 016767 000074 164036  MOV    INT1,TTY
2208 015524 004767 163626          JSR    PC,PRINT$     ;TYPE LOCATION-SUPRESS ZEROS
2209 015530 032777 001000 010070  DK11:  BIT    #89,DRPDS     ;IS DRIVE UNSAFE?
2210 015536 001401          BEQ    .+4
2211 015540 000000          HALT
2212 015542 032777 002000 010056  BIT    #810,DRPDS    ;DRIVE UNSAFE
2213 015550 001403          BEQ    INTEXT        ;SEEK INCOMPLETE?
2214 015552 112777 000015 010026  MOVB   #15,DRPCS     ;BRANCH IF COMPLETE
2215 015560 105777 010022          INTEXT: TSTB        ;RECALIBRATE
2216 015564 100375          BPL    .-4           ;WAIT FOR DONE
2217 015566 005777 010034          TST   DRPDS
2218 015572 100375          BPL    .-4           ;WAIT FOR READY
2219 015574 005767 004676          TST   BCKFLG        ;DID WE COME FROM BACKGROUND TEST?
2220 015600 001402          BEQ    1$           ;BRANCH IF NO
2221 015602 012716 022464          MOV   #NPRRET,(SP)  ;MODIFY RETURN ADDR
2222 015606 010667 166666          1$:   MOV   SP,INTFLG ;SET INTERRUPT OCCURRED FLAG
2223 015612 000002
2224
2225 015614 000000          INTO:  0
2226 015616 000000          INT1:  0
2227
2228          ;THIS ROUTINE IF SWITCH 07 OFF ISSUES A SEEK RANDOM YET NOT DISTROYING
2229          ;THE CYLINDER UNDER TEST
2230
2231 015620 105737 177570          SEEK:  TSTB        2$SWR      ;ISSUE A RANDOM SEEK?
2232 015624 100534          BMI   SEEND        ;BR IF NO
2233 015626 004567 000736          JSR   RS,DSKNOS    ;SELECT THE UNIT
2234 015632 005767 163402          TST   ERRFLG       ;ARE WE LOOPING ON ERROR?
2235 015636 001031          BNE   2$           ;BR IF YES-DO SAME SEEK AS THE ERROR
2236 015640 062767 123455 010052  1$:   ADD   #123455,SEEK1 ;GET A RANDON NUMBER
2237 015646 042767 177000 010044  BIC   #177000,SEEK1 ;CLEAR OUT UNWANTED CYLINDER BITS
2238 015654 022767 000625 010036  CMP   #625,SEEK1   ;VALID CYLINDER NUMBER
2239 015662 002766          BLT   1$           ;BR IF TOO BIG
2240 015664 016746 010030          MOV   SEEK1,-(SP)  ;PUT NEW CYLINDER ON STACK
2241 015670 166716 007750          SUB   CYLINDER,(SP);CREAT A DIFFERANCE ON STACK
2242 015674 100405          BMI   7$           ;BR IF DIFFERANCE IS -
2243 015676 021627 000200          CMP   (SP),#200   ;IS DIFFERANCE BETWEEN CYLINDER AND RANDOM CYL
2244          ;GREATER THAN 200 CYLINDERS
2245 015702 100006          BPL   8$           ;YES, CLEANUP STACK AND CONTINUE
2246 015704 005726          9$:   TST   (SP)+    ;NO, CLEAN OFF STACK
2247 015706 000754          BR    1$           ;GET NEW CYLINDER RANDOM NUMBER
2248 015710 021627 177600          7$:   CMP   (SP),#-200
2249 015714 100401          BMI   8$           ;BR IF DIFFERANCE GRATER THAN 200 CLYINDERS

```

```

2250 015716 000772          BR          9$          ;DIFFERANCE LESS THAN 200 GET ANOTHER CYL NC
2251 015720 005726          8$: TST      (SP)+        ;CLEANOFF STACK
2252 015722 017746 007654 2$: MOV      @VECTOR,-(SP) ;SAVE VECTOR
2253 015726 017746 007652 MOV      @STATUS,-(SP)  ;SAVE PRIORITY
2254 015732 012777 016120 007642 MOV      @SEKDON,@VECTOR ;SETUP TRAP FOR RANDOM SEEK
2255 015740 012777 000340 007636 MOV      #340,@STATUS
2256 015746 016777 007746 007642 3$: MOV      SEEK1,@RPCA    ;SET RANDOM CYLINDER
2257 015751 012777 177777 007644 MOV      #-1,@RPDS      ;CLEAR ATTENTIONS
2258 015762 042777 000100 007616 BIC      #86,@RPCS      ;CLEAR INTERUPT DONE
2259 015770 052777 020011 007610 BIS      #20011,@RPCS   ;SEEK INTERUPT ON ATTENTION
2260 015776 000001          WAIT
2261 016000 005777 007602          TST      @RPCS          ;ERROR?
2262 016004 100035          SPL      4$            ;BR IF NO ERROR
2263 016006 104400          HLT
2264 016010 032777 004000 007570 BIT      #811,@RPCS     ;DISK ERROR DURING RANDOM SEEK
2265 016016 001006          BNE      5$            ;SEEK INCOMPLETE?
2266 016020 032777 001000 007600 BIT      #89,@RPDS      ;BR IF SEEK INCOMPLETE
2267 016026 001402          BEQ      5$            ;FILE UNSAFE?
2268 016030 000000          HALT
2269 016032 000410          BR        6$            ;BR IF NOT UNSAFE
2270 016034          5$:
2271 016034 004567 163224          JSR      R5,PRINT$     ;PRINT MESSAGE
2272 016040 024115          MES14
2273 016042 016767 007652 163512 MOV      SEEK1,TTY
2274 016050 004767 163302          JSR      PC,PRINT$     ;TYPE LOCATION-SUPRESS ZERCS
2275 016054 012777 000015 007524 6$: MOV      #15,@RPCS     ;DO A HOME SEEK
2276 016062 105777 007520          TSTB    @RPCS
2277 016066 100375          BPL     .-4            ;WAIT FOR DONE
2278 016070 005777 007532          TST     @RPDS
2279 016074 100375          BPL     .-4            ;WAIT FOR READY
2280 016076 000723          BR      3$            ;TRY SEEK AGAIN
2281 016100 005777 007522          4$: TST      @RPDS
2282 016104 100375          BPL     .-4            ;WAIT FOR READY
2283 016106 012677 007472          MOV     (SP)+,@STATUS  ;RESTORE TRAPS
2284 016112 012677 007464          MOV     (SP)+,@VECTOR
2285 016116 000207          SEEND:  RTS      PC    ;RETURN FROM RANDOM SEEK
2286 016120 000C02          SEKDON: RTI           ;RANDOM SEEK DONE
2287
2288
2289          ;ROUTINE TO SET UP CYLINDER AND DISK ADDRESS FROM
2290          ;OPERATOR INPUTS DURING CONVERSATION MODE.
2291
2292 016122 032767 000040 007502 OPDSEL: BIT      #85,FLAG   ;USE OPERATOR ADDR?
2293 016130 001001          BNE     .+4
2294 016132 000207          RTS     PC            ;NO
2295 016134 016767 007474 007502 MOV      SCYL,CYLINDER ;GET CYLINDER ADDR
2296 016142 016767 007472 007476 MOV      SSEC,DMA      ;GET SECTOR ADDR
2297 016150 116767 007462 007471 MOVB     SHED,DMA+1    ;GET HEAD ADDR
2298 016156 000207          RTS     PC
2299
2300
2301          ;ROUTINE TO SETUP DISK BUFFERS
2302          ;ADD WORD COUNT TO STARTING DISK ADDRESSES
2303          ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
2304
2305

```



```

2306 016160 032767 000040 007444 DISBUF: BIT #B5,FLAG ;DID OPERATOR SUPPLY ADDR?
2307 016166 001401 BEQ .+4
2308 016170 000461 BR BUFEXIT ;OPERATOR DEFINED DISK ADDR
2309 016172 004767 000520 JSR PC,BLSZ ;DEFINE BLOCK SIZE
2310 016176 016767 007462 007500 MOV BLOCK,WORK1
2311 016204 016767 007436 007470 INCSEC: MOV DMA,WORK ;GET DISK ADDR
2312 016212 042767 177760 007462 SIC #177760,WORK ;MASK OUT SECTOR COUNT
2313 016220 022767 000011 007454 CMP #11,WORK ;CHECK FOR LAST SECTOR
2314 016226 001406 BEQ INCSUR ;CHECK SURFACE
2315 016230 005267 007412 INC DMA ;+1 SECTOR COUNT
2316 016234 005367 007424 DECBLK: DEC BLOCK ;-1 FROM BLOCK COUNT
2317 016240 001432 BEQ CMDAE ;CMP DMA TO RPDA
2318 016242 000760 BR INCSEC ;RECYCLE
2319 016244 042767 000017 007374 INCSUR: BIC #17,DMA ;FETCH ADDRESS
2320 016252 016767 007370 007422 MOV DMA,WORK
2321 016260 042767 160377 007414 BIC #160377,WORK
2322 016266 122767 000023 007407 CMPB #23,WORK+1
2323 016274 001403 BEQ SWSUR ;+1 SURFACE
2324 016276 105267 007345 INCB DMA+1 ;INC HEAD NUMBER
2325 016302 000754 BR DECBLK ;RECYCLE
2326 016304 005067 007336 SWSUR: CLR DMA ;CLEAR THE DISK ADDRESS
2327 016310 005267 007330 INC CYLINDER
2328 016314 022767 000626 007322 C,IP #626,CYLINDER ;HAS LAST CYL BEEN EXCEEDED?
2329 016322 001404 BEQ BUFEXIT ;BRANCH IF YES
2330 016324 000743 BR DECBLK
2331 ;COME HERE AFTER DETERMINING THE STARTING ADDR OF THE NEXT
2332 ;TRANSFER. NOW CHECK TO SEE THERE IS ENOUGH ROOM ON THE DISK
2333 ;TO MAKE THE TRANSFER. IF NOT MODIFY THE WORD COUNT FOR THE FINAL
2334 ;OUTPUT.
2335
2336
2337 016326 105767 007300 CMDAE: TSTB FLAG ;CHECK FOR LAST DISK BUFFER
2338 016332 100015 BPL BUFEXIT
2339 016334 005067 007306 BUFEXIT: CLR DMA ;CLEAR ADDRESS BITS
2340 016340 005067 007300 CLR CYLINDER ;CLR CYLINDER REGISTER
2341 016344 062716 000002 ADD #2,(6) ;INC STACK POINTER
2342 016350 042767 000200 007254 BIC #200,FLAG
2343 016356 016767 007274 007256 MOV SWRDCT,WRDCT
2344 016364 000500 BR EXTDR ;EXIT
2345 016366 005067 007314 BUFINX: CLR WORK2 ;CLEAR BLOCK COUNTER
2346 016372 016767 007250 007302 MOV DMA,WORK
2347 016400 016767 007240 007302 MOV CYLINDER,WORK3
2348 016406 042767 160360 007266 BIC #160360,WORK
2349 016414 005267 007266 XINCSEC: INC WORK2 ;INCREMENT BLOCK COUNT
2350 016420 005367 007260 DEC WORK1 ;DECREMENT TOTAL BLOCKS REQUIRED
2351 016424 001460 BEQ EXTDR ;EXIT IF BLOCK COUNT SATISFIED
2352 016426 122767 000011 007246 CMPB #11,WORK ;CHECK THE DISK ADDRESS TO
2353 016434 001403 BEQ XINCSUR ;SEE IF THERE IS ENOUGH ROOM
2354 016436 005267 007240 INC WORK ;TO HANDLE THE OUTPUT REQUESTED
2355 016442 000764 BR XINCSEC
2356 016444 105067 007232 XINCSUR: CLRB WORK
2357 016450 122767 000023 007225 CMPB #23,WORK+1
2358 016456 001403 BEQ IS
2359 016460 105267 007217 INCB WORK+1
2360 016464 000753 BR XINCSEC
2361 016466 005067 007210 IS: CLR WORK

```

```

2362 016472 022767 000625 007210      CMP      #625,WORK3      ;ARE WE ON THE LAST CYLINDER?
2363 016500 001403                      BEQ      2$            ;BRANCH IF YES
2364 016502 005267 007202                      INC      WORK3
2365 016506 000742                      BR       XINCSEC
2366 016510 016767 007172 007124 2$:    MOV      WORK2,WRDCT  ;COME HERE IF THERE IS NOT
2367 016516 000241                      CLC
2368 016520 006167 007116                      ROL      WRDCT        ;ENOUGH ROOM TO HANDLE THE
2369 016524 006167 007112                      ROL      WRDCT        ;REQUESTED OUTPUT. MODIFY THE
2370 016530 006167 007106                      ROL      WRDCT        ;WORDCOUNT TO FILL THE REMAINING
2371 016534 006167 007102                      ROL      WRDCT        ;SURFACE.
2372 016540 006167 007076                      ROL      WRDCT
2373 016544 006167 007072                      ROL      WRDCT
2374 016550 006167 007066                      ROL      WRDCT
2375 016554 006167 007062                      ROL      WRDCT
2376 016560 052767 000200 007044      BIS      #200,FLAG
2377 016566 000207                      EXTDR:  RTS      PC      ;EXIT
2378
2379
2380                      ;ROUTINE TO SELECT THE DISK UNIT
2381
2382 016570 016767 007036 007104  DSKNGS: MOV      FLAG,WORK      ;FETCH THE FLAG WORD
2383 016576 006067 007100                      ROR      WORK
2384 016602 006067 007074                      ROR      WORK
2385 016606 000241                      CLC
2386 016610 000367 007066                      SWAB     WORK
2387 016614 042767 174377 007060      BIC      #174377,WORK  ;MASK THE DISK NUMBER
2388 016622 016777 007054 006756      MOV      WORK,DRPCS   ;LOAD THE ADDRESS IN THE ADDRESS REG
2389 016630 005777 006772                      TST      DRPDS
2390 016634 100401                      BMI      1$          ;IS THE UNIT READY?
2391 016636 104400                      HLT
2392 016640 000205                      1$:      RTS      R5    ;BRANCH IF READY
2393                      ;SELECTED UNIT NOT READY
2394                      ;EXIT
2395
2396                      ;INITIALIZE THE VECTORS
2397 016642 012767 001112 161164  INIT:  MOV      #ERROR,34      ;SETUP TRAP VECTOR
2398 016650 012767 000340 161160      MOV      #PRI7,36
2399 016656 012767 001004 161144      MOV      #SCOPE$,30   ;SETUP EMT VECTOR
2400 016664 012767 000340 161140      MOV      #PRI7,32
2401 016672 012777 015224 006702      MOV      #DKINT,@VECTOR ;SETUP DISK INTERRUPT VECTOR
2402 016700 012777 000340 006676      MOV      #PRI7,@STATUS
2403 016706 012737 000340 177776      MOV      #PRI7,@PSW   ;LOCKOUT INTERRUPTS
2404 016714 000207                      RTS      PC
2405
2406                      ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
2407
2408
2409 016716 012767 000377 006740  BLSZ:  MOV      #377,BLOCK  ;DRIVE BLOCK SIZE
2410 016724 016767 006712 006750      MOV      WRDCT,WORK   ;FETCH WORD COUNT
2411 016732 036767 006726 006742      BIT      BLOCK,WORK
2412 016740 001410                      BEQ      RORBLK
2413 016742 046767 006716 006732      BIC      BLOCK,WORK   ;SET UP BLOCK OVERFLOW
2414 016750 005267 006710                      INC      BLOCK
2415 016754 066767 006704 006720      ADD      BLOCK,WORK
2416 016762 000367 006714                      RORBLK: SWAB     WORK
2417 016766 016767 006710 006670      MOV      WORK,BLOCK  ;BLOCK COUNT

```

```

2418 016774 000207          RIS      PC          ;EXIT
2419
2420
2421          ; DETERMINE THE APPROPRIATE ATTENTION BIT FROM
2422          ; THE UNIT NUMBER.
2423
2424 016776 016701 006630      GATTN:  MOV      FLAG,R1
2425 017002 006001          POR      R1
2426 017004 006001          RJR      R1          ;GET UNIT NUMBER
2427 017006 005067 000014      CLR      ATTN
2428 017012 042701 177770      BIC      #177770,R1    ;ISOLATE UNIT
2429 017016 116167 017030 000002  MOVB     ATTNB(R1),ATTN ;GET ATTENTION BIT
2430 017024 000207          RTS      PC
2431
2432
2433 017026 000000          ATTN:    0
2434 017030          001      002      004  ATTNB:  .BYTE   1,2,4,10,20,40,100,200
2435 017033          010      020      040
2436 017036          100      200
2437
2438          .EVEN
2439
2440
2441
2442          ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
2443
2444          ;ENTER FROM JSR PC PASEL
2445 017040 016700 006606      PASEL:  MOV      PATNU,RO    ;SET UP PATTERN NUMBER
2446 017044 016767 006572 006630  MOV      WRDCT,WORK    ;SET UP WORK
2447 017052 012701 025730      MOV      #OUTBUF,R1    ;LOC. OF OUT BUFFER
2448 017056 032767 000500 006546  BIT      #500,FLAG     ;BK WITH MEM. MANAG. OR NO MEM.MANAG.
2449 017064 001002          BNE      1$           ;BR IF TRUE
2450 017066 012701 040000      MOV      #40000,R1     ;START OF MEM. MANAG. BUFFER
2451 017072 022700 000034      1$:    CMP      #34,RO      ;TEST FOR RANDOM DATA NUMBER
2452 017076 001423          BEQ      RANDOM       ;GO GENERATE RANDOM DATA
2453 017100 022700 000032      CMP      #32,RO      ;IS THIS PATTERN 15
2454 017104 001406          BEQ      PATT32
2455 017106 016021 017762      FILDAT: MOV      PATO(0),(1)+ ;FILL BUFFER
2456 017112 005367 006564      DEC      WORK        ;DEC. WORK COUNT
2457 017116 001373          BNE      FILDAT      ;LOAD NEXT WORD
2458 017120 000207          RTS      PC         ;BUFFER FULL
2459 017122 012721 177777      PATT32: MOV      #177777,(1)+ ;INSERT ALL ONES PATTERN
2460 017126 005367 006550      DEC      WORK
2461 017132 001404          BEQ      1$
2462 017134 005021          CLR      (1)+        ;LOAD ZERO PATTERN
2463 017136 005367 006540      DEC      WORK        ;DECREMENT WORD COUNT
2464 017142 001367          BNE      PATT32     ;LOOP IF NOT ZERO
2465 017144 000207          1$:    RTS      PC         ;EXIT
2466          ;RANDOM DATA GENERATOR SUBROUTINE
2467 017146 016767 000134 000136  RANDOM: MOV      LONUN,LOSAV
2468 017154 016767 000130 000132  MOV      HINUN,HISAV
2469 017162 016700 000120      1$:    MOV      LONUN,RO    ;SET UP RO WITH 5 DIGITS LOW
2470 017166 016704 000116      MOV      HINUN,R4    ;SET UP R1 WITH 5 DIGITS HIGH
2471 017172 012703 000007      MOV      #7,R3      ;SET UP SHIFT COUNT
2472 017176 005002          CLR      R2         ;CLEAR R2
2473 017200 006300          2$:    ASL      RO      ;SHIFT RO LEFT AND
  
```

2474	017202	006104			RUL	R4	; ROTATE CARRY INTO LSB OF R1 INTO	
2475	017204	006102			ROL	R2	; ROTATE CARRY OUT OF R1 INTO R2	
2476	017206	005303			DEC	R3	; DECREMENT R3	
2477	017210	001373			BNE	2\$; CONTINUE SHIFT LOOP	
2478	017212	066702	000070		ADD	LONUM,R2	; ADDN IN NUMBER TO MAKE X 129	
2479	017216	005504			ADC	R4	; PROPOGATE CARRY	
2480	017220	066704	000064		ADD	HINUM,R4	; ADDN IN NUMBER TO MAKE X 129	
2481	017224	005502			ADC	R2	; PROPOGATE CARRY	
2482	017226	062700	001057		ADD	#1057,R0	; ADDN LOW CONSTANT	
2483	017232	005504			ADC	R4	; PROPOGATE CARRIES	
2484	017234	005502			ADC	R2	; PROPOGATE AGAIN	
2485	017236	062704	047401		ADD	#47401,R4	; ADDN HIGH CONSTANT	
2486	017242	005502			ADC	R2	; PROPOGATE CARRY	
2487	017244	062702	000006		ADD	#6,R2	; ADDN HIGHEST CONSTANT	
2488	017250	060200			ADD	R2,R0	; REPRIME R0 WITH HIGH DIGIT	
2489	017252	005504			ADC	R4	; PROPOGATE CARRY	
2490	017254	010067	000026		MOV	R0,LONUM	; PUT R0 BACK IN LONUM	
2491	017260	010021			MOV	R0,(1)+	; HOLD LONUM FOR PROGRAM	
2492	017262	005367	006414		DEC	WORK		
2493	017266	001406			BEQ	EXGEN		
2494	017270	010467	000014		MOV	R4,HINUM	; PUT R1 BACK IN HINUM	
2495	017274	010421			MOV	R4,(1)+	; HOLD HINUM FOR PROGRAM	
2496	017276	005367	006400		DEC	WORK		
2497	017302	001327			BNE	1\$		
2498	017304	000207			RTS	PC	; RETURN TO PROGRAM	
2499	017306	000000			LONUM:	0		
2500	017310	000000			HINUM:	0		
2501	017312	000000			LOSAY:	0		
2502	017314	000000			HISAV:	0		
2503								
2504								
2505	017316	032767	000002	161716	MSG:	BIT	#B1,HLTCT\$; TYPE ENTIRE MESSAGE
2506	017324	001103			BNE	1\$; BRANCH IF NO
2507	017326	004567	161732		JSR	R5,PRINT\$; PRINT MESSAGE
2508	017332	024002			MESB			
2509	017334	004567	161724		JSR	R5,PRINT\$; PRINT MESSAGE
2510	017340	023611			MES2A			
2511	017342	017767	006260	162212	MOV	0RPPDS,TTY		
2512	017350	004767	161770		JSR	PC,PRINTR		; TYPE LOCATION WITH LEADING ZEROS
2513	017354	004567	161704		JSR	R5,PRINT\$; PRINT MESSAGE
2514	017360	023567			MES1A			
2515	017362	017767	006236	162172	MOV	0RPER,TTY		
2516	017370	004767	161750		JSR	PC,PRINTR		; TYPE LOCATION WITH LEADING ZEROS
2517	017374	004567	161664		JSR	R5,PRINT\$; PRINT MESSAGE
2518	017400	023600			MES2			
2519	017402	017767	006200	162152	MOV	0RPCS,TTY		
2520	017410	004767	161730		JSR	PC,PRINTR		; TYPE LOCATION WITH LEADING ZEROS
2521	017414	004567	161644		JSR	R5,PRINT\$; PRINT MESSAGE
2522	017420	023622			MES2B			
2523	017422	017767	006170	162132	MOV	0RPCA,TTY		
2524	017430	004767	161710		JSR	PC,PRINTR		; TYPE LOCATION WITH LEADING ZEROS
2525	017434	004567	161624		JSR	R5,PRINT\$; PRINT MESSAGE
2526	017440	023633			MES2C			
2527	017442	017767	006152	162112	MOV	0RPDA,TTY		
2528	017450	004767	161670		JSR	PC,PRINTR		; TYPE LOCATION WITH LEADING ZEROS
2529	017454	004567	161604		JSR	R5,PRINT\$; PRINT MESSAGE

2530	017460	023644			MES20		
2531	017462	017767	006142	162072	MOV	@SUCA, TTY	
2532	017470	004767	161650		JSR	PC, PRINTR	;TYPE LOCATION WITH LEADING ZEROS
2533	017474	022767	000005	006174	CMP	#5, TESTNO	;TEST 5?
2534	017502	001404			BEQ	4\$;BR IF YES
2535	017504	022767	000006	006164	CMP	#6, TESTNO	;TEST 6?
2536	017512	001010			SNE	1\$;BR IF NO
2537	017514				4\$:		
2538	017514	004567	161544		JSR	R5, PRINT\$;PRINT MESSAGE
2539	017520	024230			MES21		;READ COUNTER =
2540	017522	016767	006174	162032	MOV	CNTA, TTY	
2541	017530	004767	161622		JSR	PC, PRINTS	;TYPE LOCATION-SUPRESS ZEROS
2542	017534	032767	000001	161500	1\$:	BIT	#B0, HLTCT\$
2543	017542	001001			BNE	2\$;TYPE EXP-REC
2544	017544	000207			RTS	PC	;BRANCH IF YES
2545	017546	032767	000002	161466	2\$:	BIT	#B1, HLTCT\$
2546	017554	001403			BEQ	3\$	
2547	017556	004567	161502		JSR	R5, PRINT\$;PRINT MESSAGE
2548	017562	024147			MES17		
2549	017564	032767	000004	161450	3\$:	BIT	#B2, HLTCT\$
2550	017572	001450			BEQ	5\$;TYPE MEMORY MANAGEMENT REGISTERS?
2551	017574	004567	161464		JSR	R5, PRINT\$;BR IF NO
2552	017600	024327			MES23		;PRINT MESSAGE
2553	017602	013767	172344	161752	MOV	@#KIPAR2, TTY	
2554	017610	004767	161542		JSR	PC, PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2555	017614	004567	161444		JSR	R5, PRINT\$;PRINT MESSAGE
2556	017620	024344			MES24		
2557	017622	013767	172346	161732	MOV	@#KIPAR3, TTY	
2558	017630	004767	161522		JSR	PC, PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2559	017634	004567	161424		JSR	R5, PRINT\$;PRINT MESSAGE
2560	017640	024367			MES25		
2561	017642	013767	172350	161712	MOV	@#KIPAR4, TTY	
2562	017650	004767	161502		JSR	PC, PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2563	017654	004567	161404		JSR	R5, PRINT\$;PRINT MESSAGE
2564	017660	024404			MES26		
2565	017662	013767	172352	161672	MOV	@#KIPAR5, TTY	
2566	017670	004767	161462		JSR	PC, PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2567	017674	004567	161364		JSR	R5, PRINT\$;PRINT MESSAGE
2568	017700	024427			MES27		
2569	017702	013767	172354	161652	MOV	@#KIPAR6, TTY	
2570	017710	004767	161442		JSR	PC, PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2571	017714				5\$:		
2572	017714	004567	161344		JSR	R5, PRINT\$;PRINT MESSAGE
2573	017720	024162			MES18		
2574	017722	016767	000030	161632	MOV	EXPS, TTY	
2575	017730	004767	161410		JSR	PC, PRINTR	;TYPE LOCATION WITH LEADING ZEROS
2576	017734	004567	161324		JSR	R5, PRINT\$;PRINT MESSAGE
2577	017740	024176			MES19		
2578	017742	016767	000012	161612	MOV	RECS, TTY	
2579	017750	004767	161370		JSR	PC, PRINTR	;TYPE LOCATION WITH LEADING ZEROS
2580	017754	000207			RTS	PC	
2581	017756	000000			EXPS:	0	
2582	017760	000000			RECS:	0	
2583							
2584							
2585							

```

017762 163126
017764 052525
017766 125252
017770 031463
017772 027417
017774 010421
017776 021042
020000 042104
020002 104210
020004 167356
020006 156735
020010 135673
020012 073567
020014 000001
020016 012767 177775 005654
020024 016767 005612 005654
020032 012767 025730 005620
020040 032767 000500 005564
020046 001003
020050 012767 040000 005602
020056 005067 005566
020062 016767 177224 161612
020070 016767 177220 161602
020076 022767 000034 005546
020104 001422
020106 022767 000032 005536
020114 001037
020116 005767 005526
020122 001006
020124 012767 177777 177624
020132 010667 005512
020136 000433
020140 005067 005504
020144 005067 177606
020150 000426
020152 005767 005472
020156 001010
020160 004767 161400
020164 016767 161512 177564
020172 010667 005452
020176 000413
020200 005067 005444
020204 016767 161470 177544
020212 000405
020214 016700 005432
020220 016067 017762 177530

```

EVEN
:RP11 DATA PATTERNS

```

PATO: 163126
PAT1: 052525
PAT2: 125252
PAT3: 031463
PAT4: 027417
PAT5: 010421
PAT6: 021042
PAT7: 042104
PAT10: 104210
PAT11: 167356
PAT12: 156735
PAT13: 135673
PAT14: 073567
PAT15: 000001
:PAT16 RANDOM DATA

```

```

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

```

```

COMPAR: MOV #3, ERRCOUNT ;ERROR RETRY COUNTER
MOV WRDCT, WORK2 ;GET THE WORD COUNT
MOV #OUTBUF, SAVE ;SET UP OUTBUFFER POINTER
BIT #500, FLAG ;BK WITH MEM. MANAG. OR NO MEM. MANAG.
BNE IS ;BR IF TRUE
MOV #40000, SAVE ;START OF MEM. MANAG. OUT BUFFER
IS: CLR SWITCH ;CLEAR RANDOM PATTERN FLAG
MOV LOSAV, LONUM ;GET RANDOM BASE NOS.
MOV HISAV, HINUM
CMP #34, PATNU ;IS THIS RANDOM PATTERN?
BEQ CMLP ;BRANCH IF YES
CMP #32, PATNU ;IS THIS SPECIAL PATTERN?
BNE CMLP1 ;BRANCH IF NO
CMLP2: TST SWITCH
BNE IS
MOV #177777, EXPS ;EXPECT ALL ONES
MOV SP, SWITCH ;SET THE FLAG
BR WRDCMP ;GO COMPARE DATA
IS: CLR SWITCH
CLR EXPS ;EXPECT ALL ZEROS
BR WRDCMP ;GO COMPARE DATA
CMLP: TST SWITCH
BNE IS
JSR PC, RANDS ;GENERATE TWO RANDOM NOS.
MOV LONUM, EXPS ;GET EVEN RANDOM WORD
MOV SP, SWITCH ;SET RANDOM PATTERN FLAG
BR WRDCMP
IS: CLR SWITCH
MOV HINUM, EXPS
BR WRDCMP
CMLP1: MOV PATNU, RO
MOV PATO(RO), EXPS

```

2642	020226	027767	005426	177522	WRDCMP:	CMP	2SAVE,EXPS		;COMPARE DATA
2643	020234	001021				BNE	WDERR		;WORD IN ERROR
2644	020236	005367	005444		WRDINC:	DEC	WORK2		;DECREMENT THE WORD COUNT
2645	020242	001415				BEG	ADAM		;EXIT ROUTINE IF ZERO
2646	020244	062767	000002	005406	BLAD1:	ADD	#2,SAVE		;UPDATE PATTERN ADDRESS
2647	020252	022767	000032	005372		CMP	#32,PATNU		
2648	020260	101362				BHI	WRDCMP		;BRANCH IF STANDARD PATTERN
2649	020262	022767	000034	005362		CMP	#34,PATNU		;IS THIS RANDOM PATTERN
2650	020270	001730				BEG	CMPLP		;BRANCH IF YES
2651	020272	000711				BR	CMPLP2		;BRANCH IF YES
2652	020274	000754				BR	WRDCMP		;COMPARE NEXT WORD
2653	020276	000207			ADAM:	RTS	PC		;EXIT THIS ROUTINE
2654	020300	005767	164174		WDERR:	TST	INTFLG		;DID INTERRUPT OCCUR YET?
2655	020304	001750				BEG	WRDCMP		;BRANCH IF NO
2656	020306	017767	005346	177444		MOV	2SAVE,RECS		;GET GOOD DATA
2657	020314	010667	160720			MOV	SP,ERRFLG		;SET ERROR FLAG
2658	020320	005767	005346			TST	RDERR		;IS THIS THE FIRST READ ERROR?
2659	020324	001404				BEG	3\$;BRANCH IF YES
2660	020326	032737	000020	177570		BIT	#84,2\$SR		;PRINT ALL RETRY ERRORS?
2661	020334	001556				BEG	1\$;BRANCH IF NO
2662	020336	032767	000100	005266	3\$:	BIT	#86,FLAG		;MEMORY MANAGEMENT?
2663	020344	001002				BNE	10\$;BR IF NO
2664	020346	104405				HLT	+5		;DATA COMPARE ERROR USING MEM. MANAG.
2665	020350	000401				BR	9\$;GET AROUND HLT+3
2666	020352	104403			10\$:	HLT	+3		;DATA COMPARE ERROR
2667	020354	005067	005304		9\$:	CLR	BLOCK		;CLEAR THE BLOCK COUNTER
2668	020360	016767	005256	005314		MOV	WRDCT,WORK		;GET THE WORD COUNT
2669	020366	166767	005314	005306		SUB	WORK2,WORK		;DETERMINE DISTANCE OF FAILURE INTO BUFFER
2670	020374	162767	000400	005300	2\$:	SUB	#400,WORK		
2671	020402	100403				BMI	8\$		
2672	020404	005267	005254			INC	BLOCK		;UPDATE BLOCK COUNT FOR EACH 400 WORDS
2673	020410	000771				BR	2\$		
2674	020412	062767	000400	005262	8\$:	ADD	#400,WORK		;RESTORE POSITIVE NUMBER
2675	020420	016767	005222	005256		MOV	DMA,WORK1		;GET HEAD AND SECTOR ADDRESS
2676	020426	016767	005212	005254		MOV	CYLINDER,WORK3		;GET CYLINDER ADDRESS
2677	020434	005767	005224		5\$:	TST	BLOCK		;IS THE BLOCK COUNT ZERO?
2678	020440	001427				BEG	7\$;BRANCH IF YES
2679	020442	005367	005216			DEC	BLOCK		;DECREMENT BLOCK COUNT
2680	020446	122767	000011	005230		CMPB	#11,WORK1		;DETERMINE THE CYLINDER, HEAD,
2681	020454	001403				BEG	4\$;AND SECTOR ADDRESSES OF THE
2682	020456	005267	005222			INC	WORK1		;COMPARE ERROR
2683	020462	000764				BR	5\$		
2684	020464	105067	005214		4\$:	CLRB	WORK1		
2685	020470	122767	000023	005207		CMPB	#23,WORK1+1		
2686	020476	001403				BEG	6\$		
2687	020500	105267	005201			INCB	WORK1+1		
2688	020504	000753				BR	5\$		
2689	020506	005067	005172		6\$:	CLR	WORK1		
2690	020512	005267	005172			INC	WORK3		
2691	020516	000746				BR	5\$		
2692	020520				7\$:				
2693	020520	004567	160540			JSR	RS,PRINTS		;PRINT MESSAGE
2694	020524	024115				MES14			;GIVE CYL ADDR
2695	020526	016767	005156	161026		MOV	WORK3,TTY		
2696	020534	004767	160616			JSR	PC,PRINTS		;TYPE LOCATION-SUPRESS ZEROS
2697	020540	005067	005130			CLR	ACNVX		

2698	020544	116767	005135	005122	MOV B	WORK1+1,ACNVX	
2699	020552	004567	160506		JSR	R5,PRINT\$;PRINT MESSAGE
2700	020556	024125			MES15		;GIVE HEAD ADDR
2701	020560	016767	005110	160774	MOV	ACNVX,TTY	
2702	020566	004767	160564		JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2703	020572	116767	005106	005074	MOV B	WORK1,ACNVX	
2704	020600	004567	160460		JSR	R5,PRINT\$;PRINT MESSAGE
2705	020604	024136			MES16		;GIVE SECTOR ADDR
2706	020606	016767	005062	160746	MOV	ACNVX,TTY	
2707	020614	004767	160536		JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2708	020620	004567	160440		JSR	R5,PRINT\$;PRINT MESSAGE
2709	020624	024015			MES9		
2710	020626	016767	005050	005040	MOV	WORK,ACNVX	;GET WORD COUNT INTO SECTOR
2711	020634	005267	005034		INC	ACNVX	
2712	020640	016767	005030	160714	MOV	ACNVX,TTY	
2713	020646	004767	160504		JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2714	020652	004567	160406		JSR	R5,PRINT\$;PRINT MESSAGE
2715	020656	024101			MES13		
2716	020660	016767	005006	160674	MOV	RDERR,TTY	
2717	020666	004767	160464		JSR	PC,PRINT\$;TYPE LOCATION-SUPRESS ZEROS
2718	020672	032737	000040	177570	1\$: BIT	#B5,#SWR	;CONTINUE COMPARING?
2719	020700	001405			BEQ	11\$;BRANCH IF NO
2720	020702	005267	004772		INC	ERCOUNT	;UPDATE ERROR COUNTER
2721	020706	001402			BEQ	11\$	
2722	020710	000167	177322		JMP	WRDINC	
2723	020714	000167	177356		11\$: JMP	ADAM	
2724							
2725							
2726							
2727							
2728							
2729							
2730							
2731							
2732							
2733	020720	032767	000100	004704	EXTMEN: BIT	#B6,FLAG	;MEMORY MANAGEMENT IN USE?
2734	020726	001002			BNE	1\$;BR IF NO
2735	020730	005037	177572		CLR	#SRD	;TURN IT OFF
2736	020734	052777	000001	004644	1\$: BIS	#B0,#RPCS	;CLEAR THE DISK
2737	020742	105777	004640		TST B	#RPCS	
2738	020746	100375			BPL	.-4	
2739	020750	012737	000340	177776	MOV	#PRI7,#PSW	;LOCK UP PRIORITY LEVELS
2740	020756	012767	021026	157020	MOV	#MAXREF,4	;SET UP I/O BUS TRAP
2741	020764	012767	000340	157014	MOV	#PRI7,6	
2742	020772	012767	017446	004660	MOV	#17446,SAVE	;SET UP FOR 4K
2743	021000	005777	004654		EXREF: TST	#SAVE	;REFERENCE MEMORY
2744	021004	022767	157446	004646	CMP	#157446,SAVE	;TEST FOR 28K
2745	021012	001001			BNE	1\$;BRANCH IF LESS THAN 28K
2746	021014	000407			BR	MAXRF1	;LAST REFERENCE MADE TO I/O REG.
2747	021016	062767	020000	004634	1\$: ADD	#20000,SAVE	;SET UP FOR NEXT MEMORY REF.
2748	021024	000765			BR	EXREF	;GO REFERENCE MEMORY
2749							
2750							
2751							
2752	021026	162767	020000	004624	MAXREF: SUB	#20000,SAVE	
2753	021034	012767	000006	156742	MAXRF1: MOV	#6,4	;RESTORE I/O BUS TRAP

;EXTENDED MEMORY EXERCISER
 ;THE PROGRAM DETERMINES HOW MUCH MEMORY
 ;IS ON THE SYSTEM THEN IT
 ;GENERATES A RANDOM BUFFER THAT SIZE
 ;AND WRITES AND WRITE CHECKS THE DATA

;ENTER HERE WHEN I/O BUS ERROR OCCURS


```

2754 021042 005067 156740 CLR 6
2755 021046 005737 000042 TST 2#42 ; UNDER MONITOR CONTROL?
2756 021052 001403 BEQ 1$ ; BRANCH IF NO
2757 021054 162767 005670 004576 SUB #3000,SAVE ; ALLOW ROOM FOR THE MONITOR
2758 021062 016767 004572 000660 1$: MOV SAVE,MMSIZ ; SAVE THE MAXIMUM MEMORY ADDRESS
2759 021070 004767 000070 JSR PC,MMSIZE ; GO SIZE WITH MEMORY MANAGEMENT
2760 021074 162767 025730 004556 MMBK: SUB #OUTBUF,SAVE ; DETERMINE THE BUFFER SIZE
2761 021102 000241 CLC
2762 021104 006067 004550 RCR SAVE ; FORM WORD COUNT
2763 021110 016767 004544 004524 MOV SAVE,WRDCT ; SAVE IT
2764 021116 042767 000001 004534 BIC #80,SAVE ; MAKE ADDRESS EVEN
2765 021124 012767 025730 004536 MOV #OUTBUF,INBUF ; START OF INPUT BUFFER
2766 021132 066767 004522 004530 ADD SAVE,INBUF
2767 021140 000241 CLC
2768 021142 042767 000377 004472 BIC #377,WRDCT ; DETERMINE MAXIMUM WORD COUNT
2769 021150 016767 004466 004500 MOV WRDCT,SWRDC
2770 021156 012706 000476 EXIT: MOV #STKPTR-2,SP
2771 021162 000205 RTS R5
2772
2773 ; THIS SUB-ROUTINE MAPS FOR MEMORY MANAGEMENT AND SETS UP
2774 ; BUFFER SIZES WITH MEMORY MANAGEMENT
2775
2776 021164 032767 000100 004440 MMSIZE: BIT #86,FLAG ; USING MEMORY MANAGEMENT?
2777 021172 001340 BNE MMBK ; BR IF NO
2778 021174 005237 177572 INC 2#SRO ; TURN MEMORY MANAGEMENT BACK ON
2779 021200 012737 021246 000004 MOV #NXM,2#ERRVEC ; SET UP FOR TRAP
2780 021206 012737 000340 000006 MOV #340,2#ERRVEC+2
2781 021214 012737 000400 172344 MOV #400,2#KIPAR2 ; SETUP FOR NEXT 4K PAGE
2782 021222 012737 177406 172304 MOV #400*256.-400+UP,RW,2#KIPDR2 ; SET KIPDR2=RW UP 400 BLOCKS
2783 021230 012700 040000 MOV #40000,R0 ; SET UP FOR TEST
2784 021234 005710 1$: TST (R0)
2785 021236 062737 000040 172344 ADD #40,2#KIPAR2 ; JUMP PAGE BY 1K
2786 021244 000773 BR 1$ ; TEST NEXT PAGE
2787 021246 022737 000400 172344 NXM: CMP #400,2#KIPAR2 ; BK MACHINE?
2788 021254 001013 BNE 1$ ; BR IF MORE THAN BK
2789 021256 052767 000400 004346 BIS #88,FLAG ; BK MACHINE WITH MEMORY MANAGEMENT
2790 021264 005037 172304 CLR 2#KIPDR2 ; SET KIPDR2 TO BE NON EXISTANT
2791 021270 012737 000006 000004 MOV #6,2#4 ; RESTORE TRAP CATCHER
2792 021276 005037 000006 CLR 2#6
2793 021302 000674 BR MMBK ; FINISH MAPPING AS IF NO MEMORY MANAGEMENT
2794 021304 013767 172344 000446 1$: MOV 2#KIPAR2,PARMAX ; PUT KIPAR2 INTO WORK LOCATION
2795 021312 032767 000340 000440 BIT #340,PARMAX ; EVEN 4K?
2796 021320 001435 BEQ 4$ ; BR IF EVEN 4K BLOCK
2797 021322 032767 000140 000430 BIT #140,PARMAX ; LAST PAGE 3K?
2798 021330 001422 BEQ 3$ ; BR IF YES
2799 021332 032767 000100 000420 BIT #100,PARMAX ; LAST PAGE 2K?
2800 021340 001407 BEQ 2$ ; BR IF YES
2801 021342 162767 000040 000410 SUB #40,PARMAX ; CREAT MAX NUMBER FOR KIPAR
2802 021350 012767 002000 000376 MOV #2000,MMSIZ ; MAX NUMBER OF WORDS FOR LAST PAGE
2803 021356 000424 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS
2804 021360 162767 000100 000372 2$: SUB #100,PARMAX ; CREAT MAX NUMBER FOR KIPAR
2805 021366 012767 004000 000360 MOV #4000,MMSIZ ; MAX NUMBER OF WORDS IN LAST PAGE
2806 021374 000415 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS
2807 021376 162767 000140 000354 3$: SUB #140,PARMAX ; CREAT MAX NUMBER FOR KIPAR
2808 021404 012767 006000 000346 MOV #6000,PARMAX ; MAX NUMBER OF WORDS IN LAST PAGE
2809 021412 000406 BR MMFIN ; CREAT BUFFER SIZES AND WORD COUNTS

```

2810	021414	162767	000200	000336	4\$:	SUB	#200,PARMAX	;CREAT MAX NUMBER FOR KIPAR
2811	021422	012767	010000	000324		MOV	#10000,MMSIZ	;MAX NUMBER OF WORDS IN LAST PAGE
2812	021430	022767	001600	000322	MMFIN:	CMP	#1600,PARMAX	;PARMAX >OR= 24K
2813	021436	003436				BLE	1\$;BR IF YES
2814	021440	012700	0C0600			MOV	#600,RO	;SET UP FOR NUMBER OF 4K PAGES
2815	021444	012701	000001			MOV	#1,R1	
2816	021450	020167	000304		2\$:	CMP	R1,PARMAX	;DO WE HAVE THE NUMBER OF PAGES
2817	021454	001404				BEQ	3\$;BR IF YES
2818	021456	062700	000200			ADD	#200,RO	;INCREMENT PAGE TEST
2819	021462	005201				INC	R1	;INCREMENT NUMBER OF PAGES
2820	021464	000771				BR	2\$;TEST MORE
2821	021466	005067	004150		3\$:	CLR	WRDCT	;CLEAR WORD COUNT
2822	021472	010167	000260			MOV	R1,PDRS	;STORE NUMBER OF KIPDRS
2823	021476	062767	010000	004136	4\$:	ADD	#10000,WRDCT	;4K BUFFER
2824	021504	005301				DEC	R1	;REDUCE NUMBER OF 4K PAGES
2825	021506	001373				BNE	4\$;BR IF MORE 4K PAGES
2826	021510	066767	000240	004124		ADD	MMSIZ,WRDCT	;ADD SIZE OF LAST PAGE
2827	021516	016767	004120	004132		MOV	WRDCT,SWRDCT	;SAVE TOTAL BUFFER SIZE
2828	021524	016767	004112	000230		MOV	WRDCT,MAXWC	;SAVE AGAIN FOR CONFORMITY
2829	021532	000414				BR	SALDR	;GO SAVE THE LOADERS
2830	021534	012767	050000	000220	1\$:	MOV	#50000,MAXWC	;20K TRANSFERS
2831	021542	012767	040000	004106		MOV	#40000,SWRDCT	;SET UP HIGHEST TRANSFER SIZE
2832	021550	066767	000200	004100		ADD	MMSIZ,SWRDCT	;ADD LAST PAGE SIZE
2833	021556	012767	000005	000172		MOV	#5,PDRS	;SET UP COUNT FOR KIPDRS
2834	021564	005767	004134		SALDR:	TST	LDRFLG	;ARE THE LOADERS ALREADY RELOCATED?
2835	021570	001017				BNE	1\$;BR IF YES
2836	021572	005037	177572			CLR	#SRO	;TURN OFF MEMORY MANAGEMENT
2837	021576	005167	004122			COM	LDRFLG	;ADJUST LOADER RELOCATED FLAG
2838	021602	016700	000142			MOV	MEMSIZ,RO	
2839	021606	042700	003777			BIC	#3777,RO	;SET UP FOR 1K WORD
2840	021612	012737	021630	000004		MOV	#1\$,#ERRVEC	;SETUP TRAP CATCHER
2841	021620	012701	034000			MOV	#34000,R1	;BK START ADDRESS FOR LOADERS
2842	021624	012021			2\$:	MOV	(R0)+,(R1)+	;MOVE LOADER
2843	021626	000776				BR	2\$	
2844	021630	012737	000006	000004	1\$:	MOV	#6,#ERRVEC	;RESTORE TRAPCATCHER
2845	021636	005037	000006			CLR	#ERRVEC+2	
2846	021642	052737	000001	177572		BIS	#80,#SRO	;TURN MEMORY MANAGEMENT BACK ON IF OFF
2847	021650	016700	000102			MOV	PDR5,RO	;PUT NUMBER OF KIPDRS INTO WORK REGISTER
2848	021654	005300				DEC	RO	;SETUP FOR PROPER NUMBER OF KIPDRS
2849	021656	001422				BEQ	3\$;BR IF NO KIPDRS NEED SETTING UP
2850	021660	012737	177406	172306		MOV	#400*256.-400+UP,RW,#KIPDR3	;SET KIPDR3=RW UP 400 BLOCKS
2851	021666	005300				DEC	RO	;ENOUGH KIPDRS?
2852	021670	001415				BEQ	3\$;BR IF ENOUGH
2853	021672	012737	177406	172310		MOV	#400*256.-400+UP,RW,#KIPDR4	;SET KIPDR4=RW UP 400 BLOCKS
2854	021700	005300				DEC	RO	;ENOUGH KIPDRS?
2855	021702	001410				BEQ	3\$;BR IF YES
2856	021704	012737	177406	172312		MOV	#400*256.-400+UP,RW,#KIPDR5	;SET KIPDR5=RW UP 400 BLOCKS
2857	021712	005300				DEC	RO	;ENOUGH KIPDRS?
2858	021714	001403				BEQ	3\$;BR IF YES
2859	021716	012737	177406	172314		MOV	#400*256.-400+UP,RW,#KIPDR6	;SET KIPDR6=RW UP 400 BLOCKS
2860	021724				3\$:			
2861	021724	004567	157352			JSR	R5,PRNTFS	;FORCE PRINT THE MESSAGE
2862	021730	024250				MES22		;LOADERS HAVE BEEN RELOCATED TO BK
2863								;TO RESTORE LOADERS START AT LOCATION
2864	021732	012767	023466	157622		MOV	#RELOAD,TTY	
2865	021740	004767	157434			JSR	PC,PRINTB	;FORCE TYPE LOCATION - SUPPRESS ZEROS

```

2866 021744 000167 177206          JMP      EXIT          ;MEMORY MANAGEMENT DONE
2867
2868 021750 000000          MEMSIZ: 0
2869 021752 000000          MEX:      0           ;CONTROLLER EXTENDED MEMORY ADDRESS
2870 021754 000000          MMSIZ: 0           ;THE LAST PAGE SIZE
2871 021756 000000          PDRS: 0           ;THE NUMBER OF PAGES IN THE BUFFER
2872 021760 000000          PARMAX: 0         ;THE MAXIMUM KIPDR VALUE
2873 021762 000000          MAXWC: 0
2874
2875          ;SUB-ROUTINE TO INITIALIZE KIPAR2-6
2876
2877 021764 016700 177766          PARINT: MOV      PDRS,RO          ;GET NUMBER OF KIPARS
2878 021770 012737 000400 172344          MOV      #400,2#KIPAR2        ;SETUP FIRST PAGE
2879 021776 005300          DEC      RO                  ;DONE?
2880 022000 001422          BEQ      1$                  ;BR IF YES
2881 022002 012737 000600 172346          MOV      #600,2#KIPAR3        ;SET UP SECOND PAGE
2882 022010 005300          DEC      RO                  ;DONE?
2883 022012 001415          BEQ      1$                  ;BR IF YES
2884 022014 012737 001000 172350          MOV      #1000,2#KIPAR4       ;SETUP THIRD PAGE
2885 022022 005300          DEC      RO                  ;DONE?
2886 022024 001410          BEQ      1$                  ;BR IF YES
2887 022026 012737 001200 172352          MOV      #1200,2#KIPARS       ;SETUP FOURTH PAGE
2888 022034 005300          DEC      RO                  ;DONE?
2889 022036 001403          BEQ      1$                  ;BR IF YES
2890 022040 012737 001400 172354          MOV      #1400,2#KIPAR6       ;SETUP FIFTH PAGE
2891 022046 062716 000004          1$:    ADD      #4,(SP)
2892 022052 000207          RTS      PC
2893
2894          ;SUB-ROUTINE TO INCREMENT EACH KIPAR USED BY 1K
2895
2896 022054 005000          PARINC: CLR      RO          ;SETUP INDEX
2897 022056 016746 177674          MOV      PDRS,-(SP)          ;GET NUMBER OF KIPARS
2898 022062 006316          ASL      (SP)                ;DOUBLE IT
2899 022064 062760 000040 172344          1$:    ADD      #40,KIPAR2(RO)  ;INCREMENT KIPAR BY 1K
2900 022072 005720          TST      (RO)+               ;BUMP INDEX
2901 022074 020016          CMP      RO,(SP)             ;DONE?
2902 022076 001372          BNE      1$                  ;BR IF NO
2903 022100 005726          TST      (SP)+               ;CLEAN UP STACK
2904 022102 000207          RTS      PC                  ;RETURN
2905
2906          ;SUB-ROUTINE TO GENERATE "MEX", "WRDCT", "BUF" USING MEMORY MANAGEMENT
2907
2908 022104 013700 172344          PARREG: MOV      2#KIPAR2,RO   ;GET THE LOW REGISTER
2909 022110 042700 171770          BIC      #171770,RO          ;CLEAR ALL BUT ADDRESS BITS 17&18
2910 022114 006300          ASL      RO                  ;PROPERLY POSITION MEX BITS
2911 022116 006300          ASL      RO
2912 022120 000300          SWAB    RO
2913 022122 010067 177624          MOV      RO,MEX              ;STORE FOR DRIVE EXTENDED MEMORY BITS
2914 022126 013700 172344          MOV      2#KIPAR2,RO        ;GET THE LOW REGISTER AGAIN

```

```

2915 022132 042700 006000      BIC      #6000,R0      ;CLEAR ADDRESS BITS 17&18
2916 022136 006200              ASR      R0          ;PROPERLY POSITION AS RPBA
2917 022140 006200              ASR      R0
2918 022142 000300              SWAB     R0
2919 022144 010067 003504      MOV      R0,BUF      ;STORE AS BUS ADDRESS
2920 022150 042767 001000 003454  BIC      #B9,FLAG    ;CLEAR LAST PAGE FLAG
2921 022156 005000              CLR      R0          ;INITIALIZE INDEX
2922 022160 016700 177572      MOV      PDRS,R0    ;GET NUMBER OF KIPARS
2923 022164 005300              DEC      R0
2924 022166 006300              ASL      R0          ;FINAL INDEX VALUE
2925 022170 016046 172344      MOV      KIPAR2(R0),-(SP) ;PUT VALUE OF LAST KIPAR ON STACK
2926 022174 022667 177560      CMP      (SP)+,PARMAX ;WORKING WITH LAST PAGE?
2927 022200 001003              BNE     1$          ;BR IF NOT LAST PAGE
2928 022202 052767 001000 003422  BIS      #B9,FLAG    ;SET LAST PAGE BIT
2929 022210 032767 000002 003414 1$: BIT      #B1,FLAG    ;OPERATOR SELECTING TRANSFER SIZE
2930 022216 001010              BNE     2$          ;BR IF OPERATOR SELECTING TRANSFER SIZE
2931 022220 032767 001000 003404  BIT      #B9,FLAG    ;LAST TRANSFER?
2932 022226 001004              BNE     2$          ;BR IF YES
2933 022230 016767 177526 003404  MOV      MAXWC,WRDCT ;SETUP WORD COUNT
2934 022236 000403              BR       3$          ;EXIT FROM SUB-ROUTINE
2935 022240 016767 003412 003374 2$: MOV      SWRDCT,WRDCT ;SETUP OPERATOR OR LAST PAGE WORD COUNT
2936 022246 000207              3$: RTS      PC      ;RETURN

```

```

2937
2938
2939
2940 ;BACKGROUND TEST FOR INTERRUPTS - WORST CASE NPRS AND BUS PATTERN
2941

```

```

2942 022250 010667 000222      NPR: MOV      SP,BCKFLG ;SET BACKGROUND FLAG
2943 022254 012767 030000 000210  MOV      #30000,NPRCNT ;SETUP TIMEOUT COUNTER
2944 022262 012701 022474      MOV      #NPR1,R1
2945 022266 005011              CLR      (R1)        ;SETUP BUS PATTERN
2946 022270 000261              SEC
2947 022272      2$:
2948 022272 106111      ROLB     (R1)
2949 022274 105421      NEGB    (R1)+
2950 022276 105441      NEGB    -(R1)
2951 022300 106111      ROLB     (R1)
2952 022302 105421      NEGB    (R1)+
2953 022304 105441      NEGB    -(R1)
2954 022306 106111      ROLB     (R1)
2955 022310 105421      NEGB    (R1)+
2956 022312 105441      NEGB    -(R1)
2957 022314 106111      ROLB     (R1)
2958 022316 105421      NEGB    (R1)+
2959 022320 105441      NEGB    -(R1)
2960 022322 106111      ROLB     (R1)
2961 022324 105421      NEGB    (R1)+
2962 022326 105441      NEGB    -(R1)
2963 022330 106111      ROLB     (R1)
2964 022332 105421      NEGB    (R1)+
2965 022334 105441      NEGB    -(R1)
2966 022336 106111      ROLB     (R1)
2967 022340 105421      NEGB    (R1)+
2968 022342 105441      NEGB    -(R1)
2969 022344 106111      ROLB     (R1)
2970 022346 105421      NEGB    (R1)+

```

2971	022350	105441		NEGB	-(R1)	
2972	022352	005201		INC	R1	
2973	022354	106111		ROLB	(R1)	
2974	022356	105421		NEGB	(R1)+	
2975	022360	105441		NEGB	-(R1)	
2976	022362	106111		ROLB	(R1)	
2977	022364	105421		NEGB	(R1)+	
2978	022366	105441		NEGB	-(R1)	
2979	022370	106111		ROLB	(R1)	
2980	022372	105421		NEGB	(R1)+	
2981	022374	105441		NEGB	-(R1)	
2982	022376	106111		ROLB	(R1)	
2983	022400	105421		NEGB	(R1)+	
2984	022402	105441		NEGB	-(R1)	
2985	022404	106111		ROLB	(R1)	
2986	022406	105421		NEGB	(R1)+	
2987	022410	105441		NEGB	-(R1)	
2988	022412	106111		ROLB	(R1)	
2989	022414	105421		NEGB	(R1)+	
2990	022416	105441		NEGB	-(R1)	
2991	022420	106111		ROLB	(R1)	
2992	022422	105421		NEGB	(R1)+	
2993	022424	105441		NEGB	-(R1)	
2994	022426	106111		ROLB	(R1)	
2995	022430	105421		NEGB	(R1)+	
2996	022432	105441		NEGB	-(R1)	
2997	022434	106111		ROLB	(R1)	
2998	022436	005301		DEC	R1	
2999	022440	103401		BCS	1\$	
3000	022442	000000		HALT		; ARITHMETIC OPERATION FAILED RUN DIAG
3001	022444	005367	000022	DEC	NPRCNT	
3002	022450	001310		BNE	2\$	
3003	022452	104400		HLT		; OPERATION TIMED OUT WAITING FOR INTERRUPT
3004	022454	004567	156604	JSR	R5, PRINT\$; PRINT MESSAGE
3005	022460	024444		TIMO		
3006	022462	000000		HALT		
3007						
3008	022464	005067	000006	NPRRET: CLR	BCKFLG	
3009	022470	000207		RTS	PC	
3010	022472	000000		NPRCNT: 0		
3011	022474	000000		NPR1: 0		
3012	022476	000000		BCKFLG: 0		
3013						
3014						
3015						; THIS TEST ALLOWS THE OPERATOR TO SPECIFY TWO CYLINDER ADDRESSES
3016						; AND THE PROGRAM WILL THEN SEEK BETWEEN THEM. THE ROUTINE DOES
3017						; NOT CHECK FOR ERRORS.
3018						
3019	022500	000005		CYLSK: RESET		
3020	022502	004567	156574	JSR	R5, PRNTF\$; FORCE PRINT THE MESSAGE
3021	022506	025302		CON17		
3022	022510	004767	157170	JSR	PC, READ\$; INPUT MESSAGE
3023	022514	022767	000131 157272	CMP	#131, INPUT\$; DOES HE WANT HEAD ALIGNMENT ROUTINE
3024	022522	001002		BNE	4\$; BR IF NO
3025	022524	000167	000424	JMP	HEAD	; GO DO HEAD ALIGNMENT ROUTINE
3026	022530			4\$:		

3027	022530	004567	156546		JSR	R5,PRNTF\$;FORCE PRINT THE MESSAGE
3028	022534	025534			CON19		
3029	022536	004767	157142		JSR	PC,READ\$;INPUT MESSAGE
3030	022542	022767	000131	157244	CMP	#131,INPUT\$;DOES HE WANT HOME SEEKS?
3031	022550	001002			BNE	1\$;BR IF NO
3032	022552	000167	000516		JMP	HOMERS	;GO DO HOME SEEK ROUTINE
3033	022556			1\$:			
3034	022556	004567	156520		JSR	R5,PRNTF\$;FORCE PRINT THE MESSAGE
3035	022562	025254			CON15		;CYLINDER A
3036	022564	004767	157114		JSR	PC,READ\$;INPUT MESSAGE
3037	022570	004767	157266		JSR	PC,PACK\$;CONVERT INPUT TO A NUMBER
3038	022574	022767	000626	157532	CMP	#626,NUM\$;IS CYL ADDR TOO HIGH?
3039	022602	101765			BLOS	1\$;BRANCH IF YES
3040	022604	016767	157524	003102	MOV	NUM\$,CYLA	;SAVE FIRST ADDR
3041	022612			2\$:			
3042	022612	004567	156464		JSR	R5,PRNTF\$;FORCE PRINT THE MESSAGE
3043	022616	025267			CON16		;CYLINDER B
3044	022620	004767	157060		JSR	PC,READ\$;INPUT MESSAGE
3045	022624	004767	157232		JSR	PC,PACK\$;CONVERT INPUT TO A NUMBER
3046	022630	022767	000626	157476	CMP	#626,NUM\$;IS CYL ADDR TOO HIGH?
3047	022636	101765			BLOS	2\$;BRANCH IF YES
3048	022640	016767	157470	003050	MOV	NUM\$,CYLB	;SAVE SECOND ADDR
3049	022646			3\$:			
3050	022646	004567	156430		JSR	R5,PRNTF\$;FORCE PRINT THE MESSAGE
3051	022652	024661			CON4		;DRIVE?
3052	022654	004767	157024		JSR	PC,READ\$;INPUT MESSAGE
3053	022660	004767	157176		JSR	PC,PACK\$;CONVERT INPUT TO A NUMBER
3054	022664	022767	000010	157442	CMP	#10,NUM\$;IS UNIT # TOO HIGH?
3055	022672	101765			BLOS	3\$;BRANCH IF YES
3056	022674	000241			CLC		
3057	022676	006167	157432		ROL	NUM\$	
3058	022702	006167	157426		ROL	NUM\$	
3059	022706	016767	157422	002716	MOV	NUM\$,FLAG	;SAVE UNIT NO.
3060	022714	004767	174056		JSR	PC,GATTN	;DETERMINE ATTENTION BIT
3061	022720	004567	173644		JSR	R5,DSKNOS	;SELECT THE UNIT
3062	022724	016777	002764	002664	MOV	CYLA,DRPCA	;LOAD THE CYLINDER ADDR
3063	022732	052777	000011	002646	BIS	#11,DRPCS	;ISSUE SEEK COMMAND
3064	022740	105777	002642		TSTB	DRPCS	;WAIT FOR READY
3065	022744	100375		20\$:			
3066	022746	036777	174054	002652	BPL	20\$	
3067	022754	001774			BIT	ATTN,DRPDS	;WAIT FOR ATTENTION
3068	022756	016777	174044	002642	BEQ	1\$	
3069	022764	005777	002616		MOV	ATTN,DRPDS	;CLEAR ATTENTION BIT
3070	022770	100022			TST	DRPCS	;ANY ERRORS?
3071	022772	104400			BPL	2\$	
3072	022774	032777	004000	002624	HLT		;ERROR AFTER SEEK COMMAND
3073	023002	001415			BIT	#B11,DRPDS	;SEEK INCOMPLETE?
3074	023004	112777	000015	002574	BEQ	2\$;BRANCH IF NO
3075	023012	105777	002570		MOV	#15,DRPCS	;ISSUE HOME COMMAND
3076	023016	100375		21\$:			;WAIT FOR READY
3077	023020	036777	174002	002600	TSTB	DRPCS	
3078	023026	001774			BPL	21\$	
3079	023030	016777	173772	002570	BIT	ATTN,DRPDS	;WAIT FOR ATTENTION BIT
3080	023036	016777	002654	002552	BEQ	3\$	
3081	023044	052777	000011	002534	MOV	ATTN,DRPDS	
3082	023052	105777	002530		MOV	CYLB,DRPCA	;LOAD CYLINDER ADDR
				2\$:			;ISSUE SEEK COMMAND
				22\$:			;WAIT FOR READY

```

3083 023056 100375          BPL      22$
3084 023060 036777 173742 002540 4$: BIT      ATTN,ARPCD ;WAIT FOR ATTENTION
3085 023066 001774          BEQ      4$
3086 023070 016777 173732 002530 MOV      ATTN,ARPCD ;CLEAR ATTENTION
3087 023076 005777 002504 TST      ARPCD ;ANY ERRORS?
3088 023102 100022          BPL      5$
3089 023104 104400          HLT
3090 023106 032777 004000 002512 BIT      #B11,ARPCD ;SEEK INCOMPLETE?
3091 023114 001415          BEQ      5$
3092 023116 112777 000015 002462 MOV      #15,ARPCD ;ISSUE HOME COMMAND
3093 023124 105777 002456 23$: TST      ARPCD ;WAIT FOR READY
3094 023130 100375          BPL      23$
3095 023132 036777 173670 002466 6$: BIT      ATTN,ARPCD ;WAIT FOR ATTENTION
3096 023140 001774          BEQ      6$
3097 023142 016777 173660 002456 MOV      ATTN,ARPCD ;CLEAR ATTENTION
3098 023150 000167 177544 5$: JMP      CYLS1
3099
3100 ;THIS ROUTINE IS USED FOR HEAD ALIGNMENT - SWITCHES 00-04 IS THE
3101 ;HEAD TO BE SELECTED, SWITCHES 08-10 IS THE DRIVE TO BE SELECTED -
3102 ;WHEN SWITCH 07 IS TOGGLED A NEW HEAD AND DRIVE WILL BE SELECTED.
3103
3104 023154 000005          HEAD: RESET
3105 023156 004567 156120 JSR      RS,PRNTF$ ;FORCE PRINT THE MESSAGE
3106 023162 025347          CON18
3107 023164 000000          HALT
3108 023166 005067 002214 CLR      CON18A ;CLEAR OUT SWITCH DEFINATIONS PRINTOUT
3109 023172 012777 000222 002416 1$: MOV      #222,ARPCD ;SETUP FOR CYLINDER 146
3110 023200 013746 177570 MOV      @#SWR,-(SP)
3111 023204 011667 002472 MOV      (SP),WORK
3112 023210 042767 177600 002464 BIC      #177600,WORK ;CLEAR OUT DRIVE NUMBER
3113 023216 000367 002460 SWAB     WORK ;PUT HEAD IN PROPER POSITION
3114 023222 016777 002454 002370 MOV      WORK,ARPCD ;LOAD HEAD NUMBER INTO DISK ADDRESS
3115 023230 012667 002446 MOV      (SP)+,WORK ;GET DRIVE NUMBER
3116 023234 042767 000377 002440 BIC      #377,WORK ;GE RID OF HEAD ADDRESS
3117 023242 052767 000011 002432 BIS      #11,WORK ;SET A SEEK GO TO DRIVE NUMBER
3118 023250 016777 002426 002330 MOV      WORK,ARPCD ;SET SEEK
3119 023256 005777 002344 TST      ARPCD
3120 023262 100375          BPL      -4 ;WAIT FOR DRIVE TO BE READY
3121 023264 105737 177570 TST      @#SWR ;NEW HEAD AND/OR DRIVE
3122 023270 100375          BPL      -4 ;BR IF NO
3123 023272 000737          BR      1$ ;GO TO NEW DRIVE AND HEAD
3124
3125
3126 ;THIS ROUTINE DOES A HOME SEEK AND THEN TRIES TO DO
3127 ;A TWO WORD READ FROM CYLINDER 000 WITH OUT A SEEK TO CYLINDER 000
3128 ;LOOPING BACK TO HOME SEEK
3129
3130 023274          HOMERS:
3131 023274 004567 156002 JSR      RS,PRNTF$ ;FORCE PRINT THE MESSAGE
3132 023300 024661          CON4 ;WHAT DRIVE?
3133 023302 004767 156376 JSR      PC,READ$ ;INPUT MESSAGE
3134 023306 004767 156550 JSR      PC,PACK$ ;CONVERT INPUT TO A NUMBER
3135 023312 022767 000010 157014 CMP      #10,NUM$ ;TOO LARGE?
3136 023320 101765          BLOS    HOMERS ;BR IF YES
3137 023322 116777 157006 002260 MOV      NUM$,ARPCD1 ;LOAD UNIT NUMBER INTO CONTROLER
3138 023330 112777 000015 002250 1$: MOV      #15,ARPCD ;ISSUE A HOME SEEK AND GO

```

```

3139 023336 105777 002244 2$: TST @RPCS ;CONTROLLER READY?
3140 023342 100375 BPL 2$ ;BR IF NO
3141 023344 005777 002256 3$: TST @RPDS ;DRIVE READY?
3142 023350 100375 BPL 3$ ;BR IF NO
3143 023252 005777 002230 TST @RPCS ;ERRORS?
3144 023356 100012 BPL 4$ ;BR IF NO ERRORS
3145 023360 104400 HLT ;ERROR WITH HOME SEEKS
3146 023362 032777 001000 002236 BIT #89,@RPDS ;FILE UNSAFE?
3147 023370 001401 BEQ 5$ ;BR IF NO
3148 023372 000000 HALT ;FILE UNSAFE
3149 023374 012767 023330 155400 5$: MOV #1$,LAD ;SETUP SCOPE LOOP
3150 023402 104000 SCOPE
3151 023404 005077 002206 4$: CLR @RPCA ;SETUP FOR CYLINDER 000
3152 023410 005077 002204 CLR @RPDA ;SETUP FOR TRACK 00, SECTOR 00
3153 023414 012777 177776 002170 MOV #-2,@RPWC ;SETUP 2 WORD TRANSFER
3154 023422 012777 025730 002164 MOV #CUTBUF,@RPBA ;INTO OUTPUT BUFFER
3155 023430 052777 000017 002150 BIS #17,@RPCS ;READ WITH NO IMPLIED SEEK
3156 023436 105777 002144 6$: TST @RPCS ;DONE?
3157 023442 100375 BPL 6$ ;BR IF NO
3158 023444 005777 002136 TST @RPCS ;ERRORS?
3159 023450 100327 BPL 1$ ;BR IF NO ERRORS TO MORE HOME SEEKS
3160 023452 104400 HLT ;HOME SEEK DID NOT RETURN TO CYLINDER 000
3161 ;FOR A READ OF ONE WORD
3162 023454 012767 023330 155320 MOV #1$,LAD ;SETUP FOR SCOPE LOOP
3163 023462 104000 SCOPE
3164 023464 000703 BR HOMERS ;TRY TEST AGAIN WITH NEW DRIVE
3165
3166
3167
3168
3169 ;THIS ROUTINE RESTORES THE LOADERS FROM BK TO HIGHEST NON MEMORY
3170 ;MANAGEMENT CORE
3171
3172 023466 012700 034000 RELOAD: MOV #34000,R0 ;START OF LOADERS
3173 023472 016701 176252 MOV MEMSIZ,R1 ;TOP OF MEMORY
3174 023476 042701 003777 BIC #3777,R1 ;MAKE IT A 1K TRANSFER
3175 023502 012737 023522 000004 MOV #1$,@ERRVEC ;SET UP FOR TRAP
3176 023510 012737 000340 000006 MOV #340,@ERRVEC+2
3177 023516 012021 2$: MOV (R0)+,(R1)+ ;RESTORE LOADER
3178 023520 000776 BR 2$
3179 023522 012737 000006 000004 1$: MOV #6,@ERRVEC ;RESTORE TRAPCATCHER
3180 023530 005037 000006 CLR @ERRVEC+2
3181 023534 005037 000176 CLR @#176
3182 023540 005067 002160 CLR LDRFLG ;SET LOADER FLAG TO LOADERS AT TOP OF MEMORY
3183 023544 012707 000176 MOV #176,PC ;FINISHED
3184
3185 ;ERROR MESSAGE HEADERS
3186
3187 .EVEN
3188 023550 005015 047105 020104 MES1: .ASCIZ <15><12>/END OF PASS /
3189 023556 043117 050040 051501
3190 023564 020123 000
3191 023567 015 051012 042520 MES1A: .ASCIZ <15><12>/RPER= /
3192 023574 036522 000040
3193
3194 023600 005015 050122 051503 MES2: .ASCIZ <15><12>/RPCS= /

```


3195	023606	020075	000				
3196							
3197	023611	015	051012	042120	MES2A:	.ASCIZ	<15><12>/RPDS= /
3198	023616	036523	000040				
3199	023622	005015	050122	040503	MES2B:	.ASCIZ	<15><12>/RPCA= /
3200	023630	020075	000				
3201	023633	015	051012	042120	MES2C:	.ASCIZ	<15><12>/RPDA= /
3202	023640	036501	000040				
3203	023644	005015	052523	040503	MES2D:	.ASCIZ	<15><12>/SUCA= /
3204	023652	020075	000				
3205							
3206	023655	015	046412	046505	MES4:	.ASCIZ	<15><12>/MEM ADDR= /
3207	023662	040440	042104	036522			
3208	023670	000040					
3209							
3210	023672	005015	040527	052111	MES5:	.ASCIZ	<15><12>/WAIT 5 SEC. AND TURN OFF PDP-11 PWR/
3211	023700	032440	051440	041505			
3212	023706	020056	047101	020104			
3213	023714	052524	047122	047440			
3214	023722	043106	050040	050104			
3215	023730	030455	020061	053520			
3216	023736	000122					
3217							
3218	023740	005015	042524	052123	MES6:	.ASCIZ	<15><12>/TEST NO /
3219	023746	047040	020117	000			
3220							
3221	023753	015	052012	052117	MES7:	.ASCIZ	<15><12>/TOT REREADS ON ERR= /
3222	023760	051040	051105	040505			
3223	023766	051504	047440	020116			
3224	023774	051105	036522	000040			
3225							
3226							
3227	024002	005015	052123	052101	MES8:	.ASCIZ	<15><12>/STAT ERR/
3228	024010	042440	051122	000			
3229							
3230	024015	015	053412	020104	MES9:	.ASCIZ	<15><12>/WD CNT INTO SECT= /
3231	024022	047103	020124	047111			
3232	024030	047524	051440	041505			
3233	024036	036524	000040				
3234							
3235	024042	005015	040520	020124	MES10:	.ASCIZ	<15><12>/PAT IN USE= /
3236	024050	047111	052440	042523			
3237	024056	020075	000				
3238							
3239	024061	015	052412	044516	MES11:	.ASCIZ	<15><12>/UNIT NO. /
3240	024066	020124	047516	020056			
3241	024074	000					
3242							
3243	024075	103	046131	000	MES12:	.ASCIZ	/CYL/
3244							
3245	024101	015	051012	040505	MES13:	.ASCIZ	<15><12>/READ NO. /
3246	024106	020104	047516	020056			
3247	024114	000					
3248							
3249	024115	015	041412	046131	MES14:	.ASCIZ	<15><12>/CYL= /
3250	024122	020075	000				

3251						
3252	024125	015	044012	040505	MES15:	.ASCIZ <15><12>/HEAD= /
3253	024132	036504	000040			
3254						
3255	024136	005015	042523	052103	MES16:	.ASCIZ <15><12>/SECT= /
3256	024144	020075	000			
3257						
3258	024147	015	041412	046517	MES17:	.ASCIZ <15><12>/COMP ERR/
3259	024154	020120	051105	000122		
3260						
3261	024162	005015	054105	042520	MES18:	.ASCIZ <15><12>/EXPECTED /
3262	024170	052103	042105	000040		
3263						
3264	024176	005015	042522	053103	MES19:	.ASCIZ <15><12>/RECVD /
3265	024204	020104	000			
3266	024207	015	047012	020117	MES20:	.ASCIZ <15><12>/NO UNITS AVAIL/
3267	024214	047125	052111	020123		
3268	024222	053101	044501	000114		
3269	024230	005015	042522	042101	MES21:	.ASCIZ <15><12>/READ CNTR = /
3270	024236	041440	052116	020122		
3271	024244	020075	000040			
3272	024250	005015	042114	051522	MES22:	.ASCII <15><12>/LDRS MOVED TO BK/
3273	024256	046440	053117	042105		
3274	024264	052040	020117	045470		
3275	024272	005015	047524	051040		.ASCIZ <15><12>/TO REST LDR START AT LOC /
3276	024300	051505	020124	042114		
3277	024306	020122	052123	051101		
3278	024314	020124	052101	046040		
3279	024322	041517	020040	000		
3280	024327	015	045412	050111	MES23:	.ASCIZ <15><12>/KIPAR2 = /
3281	024334	051101	020062	020075		
3282	024342	000040				
3283	024344	020040	020040	020040	MES24:	.ASCIZ / KIPAR3 = /
3284	024352	020040	044513	040520		
3285	024360	031522	036440	020040		
3286	024366	000				
3287	024367	015	045412	050111	MES25:	.ASCIZ <15><12>/KIPAR4 = /
3288	024374	051101	020064	020075		
3289	024402	000040				
3290	024404	020040	020040	020040	MES26:	.ASCIZ / KIPAR5 = /
3291	024412	020040	044513	040520		
3292	024420	032522	036440	020040		
3293	024426	000				
3294	024427	015	045412	050111	MES27:	.ASCIZ <15><12>/KIPAR6 = /
3295	024434	051101	020066	020075		
3296	024442	000040				
3297						
3298	024444	005015	051120	041517	TIMO:	.ASCIZ <15><12>/PROC BCKGRD TEST TIMED OUT/
3299	024452	041040	045503	051107		
3300	024460	020104	042524	052123		
3301	024466	052040	046511	042105		
3302	024474	047440	052125	000		
3303						
3304						; CONVERSATION TEXT
3305						:
3306						:

3307	024501	015	051412	042124	SPECMES:	.ASCIZ (15)(12)/STOARD WCS XFERRED= /
3308	024506	051101	020104	042127		
3309	024514	020123	043130	051105		
3310	024522	042522	036504	000040		
3311						
3312	024530	005015	040504	040524	CON1:	.ASCIZ (15)(12)/DATA TEST ONLY? (Y OR N)/
3313	024536	052040	051505	020124		
3314	024544	047117	054514	020077		
3315	024552	054450	030040	020122		
3316	024560	024516	000			
3317						
3318	024563	015	046412	046125	CON2:	.ASCIZ (15)(12)/MULTI DRIVE MODE?(Y OR N)/
3319	024570	044524	042040	044522		
3320	024576	042526	046440	042117		
3321	024604	037505	054450	047440		
3322	024612	020122	024516	000		
3323						
3324	024617	015	047012	046525	CON3:	.ASCIZ (15)(12)/NUMBER OF DRIVES 1 TO 10 OCTAL?/
3325	024624	042502	020122	043117		
3326	024632	042040	044522	042526		
3327	024640	020123	020061	047524		
3328	024646	030440	020060	041517		
3329	024654	040524	037514	000		
3330						
3331	024661	015	053412	044510	CON4:	.ASCIZ (15)(12)/WHICH DRIVE?/
3332	024666	044103	042040	044522		
3333	024674	042526	000077			
3334						
3335	024700	005015	050117	044524	CON5:	.ASCIZ (15)(12)/OPTIONAL WORD COUNT? (Y OR N)/
3336	024706	047117	046101	053440		
3337	024714	051117	020104	047503		
3338	024722	047125	037524	024040		
3339	024730	020131	051117	047040		
3340	024736	000051				
3341						
3342	024740	005015	042514	043516	CON6:	.ASCIZ (15)(12)/LENGTH? (1 TO SWRDCT)/
3343	024746	044124	020077	030450		
3344	024754	052040	020117	053523		
3345	024762	042122	052103	000051		
3346						
3347	024770	005015	052123	051101	CON7:	.ASCIZ (15)(12)/STARTING HEAD?/
3348	024776	044524	043516	044040		
3349	025004	040505	037504	000		
3350						
3351	025011	015	042012	020117	CON7A:	.ASCIZ (15)(12)/DO YOU WISH TO SELECT THE DISK TEST ADDR?(Y OR N)/
3352	025016	047531	020125	044527		
3353	025024	044123	052040	020117		
3354	025032	042523	042514	052103		
3355	025040	052040	042510	042040		
3356	025046	051511	020113	042524		
3357	025054	052123	040440	042104		
3358	025062	037522	054450	047440		
3359	025070	020122	024516	000		
3360						
3361	025075	015	051412	040524	CON7B:	.ASCIZ (15)(12)/STARTING SECT?/
3362	025102	052122	047111	020107		

3363	025110	042523	052103	000077	
3364					
3365	025116	005015	052123	051101	CON7C: .ASCIZ <15><12>/STARTING CYL?/
3366	025124	044524	043516	041440	
3367	025132	046131	000077		
3368					
3369	025136	005015	050117	040524	CON8: .ASCIZ <15><12>/OPTAL DATA PAN NO.?/
3370	025144	020114	040504	040524	
3371	025152	050040	047101	047040	
3372	025160	027117	000077		
3373					
3374	025164	005015	051127	052111	CON9: .ASCIZ <15><12>/WRITE?(Y OR N)/
3375	025172	037505	054450	047440	
3376	025200	020122	024516	000	
3377					
3378	025205	015	053412	044522	CON10: .ASCIZ <15><12>/WRITE CHECK?(Y OR N)/
3379	025212	042524	041440	042510	
3380	025220	045503	024077	020131	
3381	025226	051117	047040	000051	
3382					
3383	025234	005015	042522	042101	CON11: .ASCIZ <15><12>/READ?(Y OR N)/
3384	025242	024077	020131	051117	
3385	025250	047040	000051		
3386					
3387	025254	005015	054503	020114	CON15: .ASCIZ <15><12>/CYL "A"?/
3388	025262	040442	037442	000	
3389					
3390	025267	015	041412	046131	CON16: .ASCIZ <15><12>/CYL "B"?/
3391	025274	021040	021102	000077	
3392	025302	005015	040527	052116	CON17: .ASCIZ <15><12>/WANT HEAD ALIGN ROUTINE?(Y OR N) /
3393	025310	044040	040505	020104	
3394	025316	046101	043511	020116	
3395	025324	047522	052125	047111	
3396	025332	037505	054450	047440	
3397	025340	020122	024516	020040	
3398	025346	000			
3399	025347	015	051412	052105	CON18: .ASCII <15><12>/SET SWCH THEN PRESS CONTINUE./
3400	025354	051440	041527	020110	
3401	025362	044124	047105	050040	
3402	025370	042522	051523	041440	
3403	025376	047117	044524	052516	
3404	025404	027105			
3405					.EVEN
3406	025406	005015	053523	044103	CON18A: .ASCII <15><12>/SWCH 00-04 = HEAD/
3407	025414	030040	026460	032060	
3408	025422	036440	044040	040505	
3409	025430	104			
3410	025431	015	051412	041527	.ASCII <15><12>/SWCH 08-10 = DRIVE/
3411	025436	020110	034060	030455	
3412	025444	020060	020075	051104	
3413	025452	053111	105		
3414	025455	015	052012	043517	.ASCIZ <15><12>'TOGGLE SW 07 TO SELECT NEW HEAD AND/OR DRIVE'
3415	025462	046107	020105	053523	
3416	025470	030040	020067	047524	
3417	025476	051440	046105	041505	
3418	025504	020124	042516	020127	

3419	025512	0425:0	042101	040440	
3420	025520	042116	047457	020122	
3421	025526	051104	053111	000105	
3422	025534	005015	040527	052116	CON19: .ASCIZ <15><12>/WANT CONTINUOUS HOME SEEKS? /
3423	025542	041440	047117	044524	
3424	025550	052516	052517	020123	
3425	025556	047510	042515	051440	
3426	025564	042505	051513	020077	
3427	025572	000			
3428					
3429	025573	015	042412	042116	END: .ASCIZ <15><12>/END/
3430	025600	020			
3431		025602			.EVEN

3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487

025602 000254
025604 000256
025606 176714
025610 176715
025612 176716
025614 176720
025616 176722
025620 176724
025622 176725
025624 176712
025626 176710
025630 176734

025632 000000
025634 000000
025636 000000
025640 000000
025642 000000
025644 000000
025646 000000
025650 000000
025652 000000
025654 000000
025656 000000
025660 000000
025662 000000
025664 000000
025666 000000
025670 000000
025672 000000
025674 000000
025676 000000
025700 000000
025702 000000
025704 000000
025706 000000
025710 000000
025712 000000
025714 000000
025716 000000
025720 000000
025722 000000
025724 000000

;
;
;
;
;

;DISK I/O REGISTERS

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

;DISK INTERRUPT TRAP LOCATION
;INTERUPT PRIORITY ASSIGNMENT
;DISK CONTROL REGISTER
;UPPER BYTE OF CONTROL REGISTER
;WORD COUNT REGISTER
;CURRENT ADDR REGISTER
;CYLINDER ADDR REGISTER
;DISK ADDR REGISTER
: TRACK ADDRESS
;ERROR REGISTER
;DEVICE STATUS REGISTER
;SELECTED UNIT CYLINDER ADDR REG

;DEDICATED REGISTERS

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

;INTERNAL PROGRAM FLAG WORD
;OPERATOR SELECTED CYLINDER
;OPERATOR SELECTED HEAD
;OPERATOR SELECTED SECTOR
;WORKING WORD COUNT
;WORKING CYLINDER ADDR
;WORKING DISK ADDR

;DATA PATTERN INDEX
;WORKING DATA BUFFER
;STANDARD WORD COUNT

;MAXIMUM UNIT NUMBER

;CONTAINS START OF INPUT BUFFER
;READ RETRY COUNTER

;WHEN =0 LOADERS ARE AT TOP OF MEMORY

.EVEN

3488	025726	005015				HEADER: .ASCII <15><12>
3489	025730	042115	030455	026461		OUTBUF: .ASCII 'MD-11-DZRPB-C DISK RELIABILITY TEST'
3490	025736	055104	050122	026502		
3491	025744	020103	020040	044504		
3492	025752	045523	051040	046105		
3493	025760	040511	044502	044514		
3494	025766	054524	052040	051505		
3495	025774	124				
3496	025775	015	005012	053523		.ASCII <15><12><12>'SWITCH OPTIONS'
3497	026002	052111	044103	047440		
3498	026010	052120	047511	051516		
3499	026016	005015	053523	036040		.ASCII <15><12>'SW <15> =1....HALT ON ERROR'
3500	026024	032461	020076	030475		
3501	026032	027056	027056	040510		
3502	026040	052114	047440	020116		
3503	026046	051105	047522	122		
3504	026053	015	051412	020127		.ASCII <15><12>'SW <14> =1....LOOP ON ERROR'
3505	026060	030474	037064	036440		
3506	026066	027061	027056	046056		
3507	026074	047517	020120	047117		
3508	026102	042440	051122	051117		
3509	026110	005015	053523	036040		.ASCII <15><12>'SW <13> =1....INHIBIT PRINTOUTS'
3510	026116	031461	020076	030475		
3511	026124	027056	027056	047111		
3512	026132	044510	044502	020124		
3513	026140	051120	047111	047524		
3514	026146	052125	123			
3515	026151	015	051412	020127		.ASCII <15><12>'SW <12> =1....INHIBIT BACKGROUND TEST'
3516	026156	030474	037062	036440		
3517	026164	027061	027056	044456		
3518	026172	044116	041111	052111		
3519	026200	041040	041501	051113		
3520	026206	052517	042116	052040		
3521	026214	051505	124			
3522	026217	015	051412	020127		.ASCII <15><12>'SW <11> =1....RING BELL ON ERROR'
3523	026224	030474	037061	036440		
3524	026232	027061	027056	051056		
3525	026240	047111	020107	042502		
3526	026246	046114	047440	020116		
3527	026254	051105	047522	122		
3528	026261	015	051412	020127		.ASCII <15><12>'SW <10> =1....LOOP ON TEST'
3529	026266	030474	037060	036440		
3530	026274	027061	027056	046056		
3531	026302	047517	020120	047117		
3532	026310	052040	051505	124		
3533	026315	015	051412	020127		.ASCII <15><12>'SW <09> =1....INHIBIT DATA COMPARISION'
3534	026322	030074	037071	036440		
3535	026330	027061	027056	044456		
3536	026336	044116	041111	052111		
3537	026344	042040	052101	020101		
3538	026352	047503	050115	051101		
3539	026360	051511	047511	116		
3540	026365	015	051412	020127		.ASCII <15><12>'SW <08> =1....ENTER CONVERSATION MODE'
3541	026372	030074	037070	036440		
3542	026400	027061	027056	042456		
3543	026406	052116	051105	041440		

3544	026414	047117	042526	051522
3545	026422	052101	047511	020116
3546	026430	047515	042504	
3547	026434	005015	020040	020040
3548	026442	020040	020040	020040
3549	026450	020040	020040	042524
3550	026456	052123	051440	046105
3551	026464	041505	042524	020104
3552	026472	020060	020055	020062
3553	026500	044507	042526	020123
3554	026506	042101	051104	051505
3555	026514	020123	047503	053116
3556	026522	051105	040523	044524
3557	026530	047117		
3558	026532	005015	020040	020040
3559	026540	020040	020040	020040
3560	026546	020040	020040	042524
3561	026554	052123	051440	046105
3562	026562	041505	042524	020104
3563	026570	020063	020055	020067
3564	026576	044507	042526	020123
3565	026604	040504	040524	041440
3566	026612	047117	042526	051522
3567	026620	052101	047511	116
3568	026625	015	051412	020127
3569	026632	030074	037067	036440
3570	026640	027061	027056	044456
3571	026646	044116	041111	052111
3572	026654	051040	047101	047504
3573	026662	020115	042523	045505
3574	026670	020123	052504	044522
3575	026676	043516	042040	052101
3576	026704	020101	042524	052123
3577	026712	005015	053523	036040
3578	026720	033060	020076	030475
3579	026726	027056	027056	047111
3580	026734	044510	044502	020124
3581	026742	051525	020105	043117
3582	026750	046440	046505	051117
3583	026756	020131	040515	040516
3584	026764	042507	042515	052116
3585	026772	005015	053523	036040
3586	027000	032460	027076	027056
3587	027006	027056	027056	047503
3588	027014	050115	051101	051511
3589	027022	047511	020116	051105
3590	027030	047522	051522	050040
3591	027036	044522	052116	052517
3592	027044	124		
3593	027045	015	051412	020127
3594	027052	030074	037064	027056
3595	027060	027056	027056	051056
3596	027066	051105	040505	020104
3597	027074	047111	047506	046522
3598	027102	052101	047511	116
3599	027107	015	051412	020127

.ASCII <15><12>'

TEST SELECTED 0 - 2 GIVES ADDRESS CONVERSATION'

.ASCII <15><12>'

TEST SELECTED 3 - 7 GIVES DATA CONVERSATION'

.ASCII <15><12>'SW <07> =1....INHIBIT RANDOM SEEKS DURING DATA TEST'

.ASCII <15><12>'SW <06> =1....INHIBIT USE OF MEMORY MANAGEMENT'

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

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

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

3600	027114	030074	037063	036440
3601	027122	027061	027056	051456
3602	027130	046105	041505	020124
3603	027136	042524	052123	044440
3604	027144	020116	053523	044103
3605	027152	030040	020060	047524
3606	027160	030040	000062	
3607				
3608		000001		

.END

ACNVX	025674	601*	602*	603*	604*	605	2697*	2698*	2701	2703*	2706	2710*	2711*	2712
ADAM	020276	3472#												
ADERC	005432	2645	2653#	2723										
ADERC1	005450	865	877#											
ADERR	005442	678	880#											
ADHGT	005356	869	879#											
ADHGT1	005362	857	860	862#										
ADTER1	005472	863#	876											
ADTST	003546	861	874	883	886#									
ADT1	004024	435	576#	2040										
ADT2	004502	615	618	637#										
ADT3	005564	619	716	736#										
ADT32	006012	620	899	909#										
ASKWC	003052	947#	959	976	981									
ATTN	017026	474	486#											
ATTNB	017030	683	2427*	2429*	2433#	3066	3069	3077	3079	3084	3086	3095	3097	
BCKFLG	022476	2429	2434#											
BELL	001250	2219	2942*	3008*	3012#									
BLAD1	020244	196	217#											
BLOCK	025664	2646#												
BLSZ	016716	2310	2316*	2409*	2411	2413	2414*	2415	2417*	2667*	2672*	2677	2679*	3468#
BUF	025654	2309	2409#											
BUFEXI	016334	769*	920*	947*	1008*	1155*	1253*	1382*	1444*	1549*	1590*	1631*	1728*	1909*
BUFINX	016366	1912*	2067*	2146	2919*	3464#								
B0	= 000001	2308	2329	2339#										
B1	= 000002	2338	2345#											
B10	= 002000	71#	1318	2180	2542	2736	2764	2846						
B11	= 004000	70#	505	1183	2169	2505	2545	2929						
B12	= 010000	61#	452	614	662	715	898	982	1230	1405	1678	1862	2006	2212
B13	= 020000	60#	193	458	589	597	696	830	887	934	962	1365	2018	2264
B14	= 040000	3072	3090											
B15	= 100000	59#	564	1306	1735	1749	1758	1776	1916	1925	1963			
B2	= 000004	58#	197	223	241	244	571	1744						
B3	= 000010	57#	157	557	859	1120	1276	1310	1467	1521	1558	1729	1970	
B4	= 000020	56#	210	547	701	709	835	892	939	967	975	1077	1187	1236
B5	= 000040	1314	1350	1370	1562	2076								
B6	= 000100	69#	1554	2549										
B7	= 000200	68#	607	1052										
B8	= 000400	67#	1600	1613	1640	1658	2166	2660						
B9	= 001000	66#	512	1623	1644	1668	1851	2292	2306	2718				
C	= 000001	65#	407	420	1430	1545	1682	2258	2662	2733	2776			
CHKMOD	003536	64#												
CLRTST	010346	63#	432	1487	2789									
CMDAE	016326	62#	1541	1774	1842	1968	2209	2266	2920	2928	2931	3146		
CMPLP	020152	21#												
CMPLP1	020214	570	572#											
CMPLP2	020116	1378#	1394											
CNTA	025722	2317	2337#											
COMPAR	020016	2620	2631#	2650										
CON1	024530	2622	2640#											
CON10	025205	2623#	2651											
CON11	025234	431*	1707*	1767*	1828*	1837*	1838	1932*	1990*	1999*	2000	2540	3483#	

CON15	025254	3035	3387*											
CON16	025267	3043	3390*											
CON17	025302	3021	3392*											
CON18	025347	3106	3399*											
CON18A	025406	3108*	3406*											
CON19	025534	3028	3422*											
CON2	024563	454	3318*											
CON3	024617	461	3324*											
CON4	024661	477	3051	3132	3331*									
CON5	024700	488	3335*											
CON6	024740	494	3342*											
CON7	024770	523	3347*											
CON7A	025011	508	3351*											
CON7B	025075	531	3361*											
CON7C	025116	515	3365*											
CON8	025136	540	3369*											
CON9	025164	553	3374*											
CYLA	025714	670*	671*	672*	673*	755*	756*	757*	758*	1297*	1298*	1299*	1300*	3040*
		3062	3480*											
CYLB	025716	3048*	3080	3481*										
CYLIND	025644	428*	645*	650	697	689	708*	711*	771*	774	819	864	877	918*
		921	930	932*	943*	956	958	974*	979*	1004*	1153*	1170*	1193*	1219*
		1251*	1380*	1446*	1499*	1713*	1900*	2017*	2064*	2145	2241	2295*	2327*	2328
		2340*	2347	2676	3460*									
CYLSK	022500	441	3019*											
CYLSI	022720	3061*	3098											
DATAT	012272	616	623	1679	1702*									
DATP	012430	1720	1725*	1853										
DATTES	003004	457	475*	481										
DECBK	016234	2316*	2325	2330										
DECCY	015416	2187	2191*											
DECTK	015372	2183	2186*											
DELMES	015322	2170	2177*											
DISBUF	016160	839	896	1742	1756	1847	2306*							
DISPLA=	177570	50*												
DKINT	015224	1717	2159*	2401										
DKII	015530	2167	2209*											
DMA	025646	429*	770*	776	779	781*	783*	807	868	873	875*	879	886*	917*
		1005*	1152*	1171*	1172*	1192*	1220*	1252*	1379*	1445*	1498*	1714*	1908*	2016*
		2065*	2144	2296*	2297*	2311	2315*	2319*	2320	2324*	2326*	2339*	2346	2675
		3461*												
DREAD	012616	1745	1758*											
DSKDR	002712	459*	465	468										
DSKNOR	025662	427*	469*	470*	472*	473*	578*	580	583*	585*	587*	588*	590	2022
		3467*												
DSKNOS	016570	644	742	916	1003	1344	1378	2057	2143	2233	2382*	3061		
DSKRO	012676	1771*	1817	1823										
EMTVEC=	000030	40*												
END	025573	2123	3429*											
EOPST	007110	1152*	1210											
ERCOUN	025700	2610*	2720*	3474*										
ERRFLG	001240	159	161*	210*	213*	1811	1973	2159*	2163*	2175*	2234	2657*		
ERROR	001112	190*	2397											
ERRORS	001246	211*	216*											
ERRPC	001252	200	218*											
ERRVEC=	000004	35*	409*	410*	421*	422*	2779*	2780*	2840*	2844*	2845*	3175*	3176*	3179*

.SAVE	126*	165
.SCOPE	126*	149
.SETUP	1*	
.SAPHI	1*	
.SACTI	1*	
.SAPT8	1*	
.SAPTH	1*	
.SAPTY	1*	
.SASTA	1*	
.SCATC	1*	
.SCMTA	1*	
.SCB20	1*	
.SDB20	1*	
.SDIV	1*	
.SEOP	1*	
.SERRO	1*	
.SERRT	1*	
.SMULT	1*	
.SPOWE	1*	
.SRAND	1*	
.SRDE	1*	
.SROOC	1*	
.SREAD	1*	
.SR2AZ	1*	
.SSAVE	1*	
.SSB20	1*	
.SSB20	1*	
.SSCOP	1*	
.SSIZE	1*	
.SSLPR	1*	
.STRAP	1*	
.STYPB	1*	
.STYPD	1*	
.STYPE	1*	
.STYPC	1*	
.S40CA	1*	
.1170	1*	

ROR	2985	2988	2991	2994	2997										
RTI	364	365	602	603	1398	1399	1435	1711	2095	2096	2383	2384	2425	2426	2762
RTS	162	164	212	721	2223	2286									
	226	235	243	246	279	308	335	384	392	2153	2285	2294	2298	2377	2392
	2404	2418	2430	2458	2465	2498	2544	2580	2653	2771	2892	2904	2936	3009	
SEC	2946														
SUB	202	466	1433	1806	1961	2241	2669	2670	2752	2757	2760	2801	2804	2807	2810
SWAB	363	368	373	1400	2097	2386	2416	2912	2918	3113					
TRAP	97														
TST	159	190	207	386	411	464	497	576	581	590	660	668	674	678	693
	705	753	759	763	790	794	801	820	845	856	866	870	926	952	971
	980	1012	1019	1027	1035	1039	1045	1060	1066	1099	1108	1117	1131	1158	1165
	1176	1198	1203	1263	1272	1301	1333	1360	1403	1441	1451	1464	1474	1495	1504
	1518	1520	1539	1573	1596	1609	1636	1654	1788	1811	1821	1824	1854	1973	1983
	1986	2070	2100	2102	2160	2164	2217	2219	2234	2246	2251	2261	2278	2281	2389
	2623	2631	2654	2658	2677	2743	2755	2784	2834	2900	2903	3069	3087	3119	3141
	3143	3158													
TSTB	227	232	237	264	267	316	327	348	656	699	747	788	833	843	854
	890	927	937	950	965	1017	1025	1050	1097	1106	1115	1163	1181	1196	1228
	1261	1270	1295	1331	1348	1354	1358	1368	1389	1449	1462	1502	1516	1552	1594
	1607	1634	1652	1809	1819	1981	2068	2074	2085	2215	2231	2276	2337	2737	3064
	3075	3082	3093	3121	3139	3156									
WAIT	1739	1753	1780	1920	1929	1967	2260								
.ABS	4														
.ASCII	3272	3399	3406	3410	3488	3489	3496	3499	3504	3509	3515	3522	3528	3533	3540
	3547	3558	3568	3577	3585	3593									
.ASCIZ	217	218	338	339	3188	3191	3194	3197	3199	3201	3203	3206	3210	3218	3221
	3227	3230	3235	3239	3243	3245	3249	3252	3255	3258	3261	3264	3266	3269	3275
	3280	3283	3287	3290	3294	3298	3307	3312	3318	3324	3331	3335	3342	3347	3351
	3361	3365	3369	3374	3378	3383	3387	3390	3392	3414	3422	3429	3599		
.BLKW	280	337													
.BYTE	2434														
.ENABL	1	397													
.END	3608														
.ENDC	207														
.EVEN	220	2437	2586	3187	3405	3431	3487								
.IF	206														
.LIST	1	2	143	397											
.MACR	95	126	127	128	129	130	131	132	133	134	135	137	138	139	140
	141														
.MACRO	1	136	2938												
.NLIST	1	3	143	397											
.PAGE	397	1408	2043	2138											
.REPT	1	143	1790	1945	2947	2973									
.SBTTL	629	724	901	986	1408	1689	1865	2043	2138						
.TITLE	5	397													
.WORD	785	853	923	949	1016	1024	1049	1096	1105	1114	1162	1190	1195	1223	1260
	1269	1292	1330	1384	1448	1461	1501	1515	1551	1592	1606	1633	1651	1734	1748
	1773	1915	1924	1937	2073	2084									

 ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

RP11C RELIABILITY TEST MACY11 27(732) 04-NOV-76 14:18 PAGE 93
DZRPB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

*.DZRPB.SEQ/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZRPB.P:1
RUN-TIME: 37 51 6 SECONDS
RUN-TIME RATIO: 172/95=1.8
CORE USED: 41K (81 PAGES)

