

RH11-RS03

RH11 RS03/LA RS04 DATA RLBTY
MD-11-DZRSC-E

EP DZRSC DL E

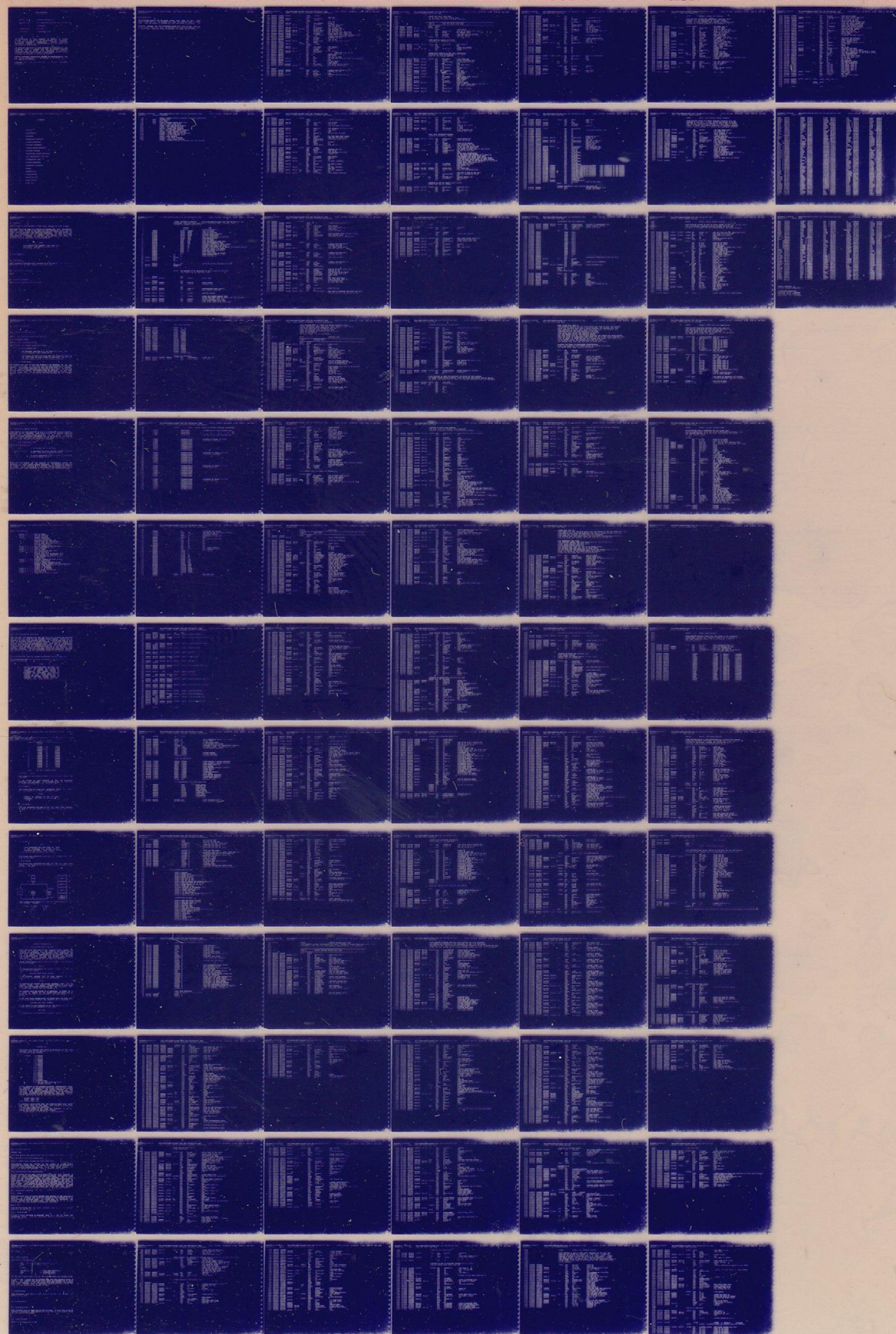
COPYRIGHT 73-76

FICHE 1 OF 1

FEB 1979

digital

MADE IN USA



IDENTIFICATION

SEQ 0001

PRODUCT CODE: MAINDEC-11-DZRS-C-E-D
PRODUCT NAME: RH11-RS03-RS03/LA-RS04 DATA RELIABILITY
DIAGNOSTIC
PROGRAM DATE: AUG 1976
MAINTAINER: DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1974, 1975, 1976 BY DIGITAL EQUIPMENT CORPORATION

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 PRELIMINARY PROGRAMS
- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
- 5. OPERATIONAL SWITCH SETTINGS
 - 5.1 DATA RELIABILITY TEST MODE
 - 5.2 CONVERSATION MODE
 - 5.3 ROUTINE ABSTRACTS
 - 5.4 SUBROUTINE ABSTRACTS
- 6. ERRORS
- 7. RESTRICTIONS
- 8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 POWER FAIL

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03, RS03/LA AND RS04 DRIVES.

THE DZRSB DISK DATA TEST IS A SERIES OF ADDRESS AND DATA RELIABILITY ROUTINES WHICH VERIFY TO THE USER THAT THE CONTROLLER (RH11) AND THE DISKS (RS03/LA OR RS04) ARE OPERATING CORRECTLY. THIS TEST SHOULD BE USED IN CONJUNCTION WITH THE DZRSB DIAGNOSTIC. IF THERE IS A POWER FAIL WHILE THE DIAGNOSTIC IS RUNNING, THE PROGRAM WILL WAIT FOR APPROXIMATELY 5 MINUTES, TO GIVE ALL THE DRIVES TIME TO COME BACK UP TO SPEED, BEFORE RESTARTING THE TEST SEQUENCE.

NOTE

THIS PROGRAM WILL DESTROY ALL DATA ON THE DISKS. TURN OFF ALL DRIVES THAT YOU DO NOT WANT TO TEST.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OF 8K OF MEMORY, AND AN RH11 CONTROLLER WITH AN RS03, RS03/LA OR AN RS04 DISK.

2.2 PRELIMINARY PROGRAMS

DZRSB

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5. (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.). ALL DOWN FOR WORST CASE (IF SWITCH-LESS CPV SIMPLY PRESS START)
- B. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- C. PRESS START.

THE PROGRAM WILL NOW MAP THE DATA BUFFERS IN 4K SEGMENTS ON -A- AND -B- PORTS UP TO 28K. IT WILL THEN TYPE OUT THE PARAMETERS OF THE DATA BUFFERS. THE PROGRAM WILL ONLY DO THIS THE FIRST TIME IT IS STARTED, FOR IT STORES THESE ADDRESSES AND CONTINUES USING THEM. TO HAVE THE PROGRAM REMAP THE SYSTEM, THE PROGRAM MUST BE RELOADED. IF YOU WISH TO GET ABOVE 28K, YOU CAN ENTER CONVERSATION MODE AND PUT THE DATA BUFFERS WHERE YOU WISH. THE SIZE OF THE DATA BUFFERS CAN NOT EXCEED 24K.

5. OPERATIONAL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC.176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED.

(I,E) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE.
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.
3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (I.E.) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

SWITCH SETTINGS ARE:

SW<15> = 1 HALT ON ERROR
SW<14> = 1 LOOP ON FUNCTION
SW<13> = 1 INHIBIT PRINTOUT
SW<12> = 1 INHIBIT COMPARISON
WITH THIS SWITCH SET, THE
PROGRAM WILL NOT COMPARE THE
DATA IT READ FROM THE DISK WITH
THE KNOWN GOOD DATA.
SW<11> = 1 HALT ON COMPLETION OF TRANSFER
SW<10> = 1 ENTER CONVERSATION MODE
SW<09> = 1 LOOP ON ERROR
SW<08> = 1 DATA RELIABILITY TEST MODE
SW<07> = 1 WAIT IN WAIT MODE
PROGRAM RUNS IN A BACKGROUND TEST
WHILE WAITING FOR INTERRUPT, WITH
SW SET PROGRAM WAITS IN A WAIT
INSTRUCTION.
SW<06> = 1 OPTIONAL TIMEOUT OF RETRY ERRORS
SW<05> = 1 INHIBIT PASS COUNT
SW<04> = 1 ALLOWS 0 ERROR TIMEOUTS IN THE
COMPARE ROUTINE BEFORE EXECUTING NEXT READ
COMMAND. WHEN SWITCH IS 0, ONLY 1 ERROR
TIMEOUT IS RECORDED.
SW<03> = 1 TIMEOUT 0 OF ERRORS
SW<02> = 1 INHIBIT MEMORY MANAGEMENT
SW<01> = 1 DATA TEST ONLY
SW<00> = 1 DROPS DRIVE AFTER 20 ERRORS

5.1 DATA RELIABILITY TEST MODE

WITH SW8 SET, THE PROGRAM WILL SET THE "BAI" BIT IN RHCS2 AND TRANSFER 64K OF DATA AT A TIME FOR ALL PATTERNS EXCEPT RANDOM. RANDOM WILL BE EXECUTED AS USUAL WITH STANDARD BUFFERS. NO COMPARES ARE DONE IN THIS MODE OF OPERATION EXCEPT ON RANDOM PATTERNS. THIS OPTION SHOULD ONLY BE USED IN DATA TEST OR CONVERSATION MODE. WHEN USED IN CONVERSATION MODE IT OVER RIDES THE NON STANDARD WORD COUNT. YOU SHOULD NOT SELECT A DESIRED DISK ADDRESS IN CONVERSATION MODE FOR IT CAN PRODUCE A DISK ADDRESS OVERFLOW ERROR FOR THIS DATA RELIABILITY TEST MODE ONLY DOES 64K WORD TRANSFERS. IF SW8 IS CHANGED, WHILE THE PROGRAM IS RUNNING, THE PROGRAM WILL FINISH ITS PASS BEFORE EXECUTING THE SWITCH CHANGE.

5.2 CONVERSATION MODE FOR PROGRAM PARAMETERS FOR DATA TEST ONLY

IN CONVERSATION MODE THE OPERATOR CAN SPECIFY ANY ONE OR ALL OF THE PROGRAM PARAMETERS.

NOTE

ONCE IN CONVERSATION MODE, THE ONLY WAY TO REMAP THE SYSTEM IS TO RELOAD THE PROGRAM. TO RESTART THE PROGRAM IN CONVERSATION MODE WITHOUT HAVING TO REANSWER THE QUESTIONS, THE STARTING ADDRESS IS 210. RESET SWITCH 10. TO RESTART THE PROGRAM WITHOUT HAVING TO REANSWER THE PORT SIZING QUESTIONS, RESTART AT 220. RESET SWITCH 10.

THE PROGRAM WILL NOW ASK SEVERAL QUESTIONS, THE TABLE BELOW WILL HELP YOU ANSWER THE QUESTIONS.

TYPE TO START AT		TYPE TO START AT	
0	000000		
1	020000	20	400000
2	040000	21	420000
3	060000	22	440000
4	080000	23	460000
5	100000	24	480000
6	120000	25	500000
7	140000	26	520000
8	160000	27	540000
9	180000	28	560000
10	200000	29	580000
11	220000	30	600000
12	240000	31	620000
13	260000	32	640000
14	280000	33	660000
15	300000	34	680000
16	320000	35	700000
17	340000	36	720000
	360000		740000

NOTE: TYPE ONLY NUMBERS SHOWN!!!

1. -A- PORT? (Y OR N)

THIS GIVES YOU THE OPTION TO TEST -A- OR -B- PORT USING THE DATA TEST.

IF THE ANSWER TO THIS QUESTION IS YES, THE FOLLOWING QUESTIONS WILL BE ASKED. IF THE ANSWER IS NO, -B- PORT WILL BE TESTED AND QUESTIONS 4 AND 5 WILL BE ASKED.

2. 1ST 4K BANK #

THIS NUMBER THAT IS TYPED WILL DETERMINE WHERE THE BUFFER AREA WILL START ON -A- PORT. USE TABLE ABOVE

NOTE:

PROGRAM IS LOCATED IN 1ST 4K BANK. THEREFORE, THIS BANK CAN NOT BE USED AS A BUFFER.

3. # OF 4K BANKS ?

THIS WILL DETERMINE THE SIZE OF THE -A- PORT DATA BUFFER. THE SIZE OF THE DATA BUFFER CAN NOT EXCEED 24K AND MUST BE IN "OCTAL".

4. 1ST 4K BANK 0

NOTE

THIS DIAGNOSTIC WILL ONLY TEST -B- PORT
IF THE PROCESSOR HAS ACCESS TO THAT
MEMORY ON -B- PORT. THIS MEMORY MUST
HAVE THE SAME ADDRESS ON ALL PORTS.

THIS NUMBER WILL DETERMINE WHERE THE DATA BUFFER AREA WILL
START ON -B- PORT.

5. # OF 4K BANKS?

THIS NUMBER WILL DETERMINE THE SIZE OF THE -B- PORT DATA
BUFFER. THE SIZE OF THE DATA BUFFER CAN NOT EXCEED 24K AND
MUST BE IN "OCTAL".

EXAMPLE:

```

12K XXXXX
  X  X
  XMEMX
  X  X
0K XXXXX
  X
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
X
X
XXXXXXX
X CPU X
X CPU X
XXXXXXX
X
X
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
12K XXXXX
  XMEMX
  XXXXX
  X
X-A-PORT
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXX 16K
X X
X BANK 3 X
X X
XXXXXXXXXXXXXXXXXXXX 12K
X X
X BANK 2 X
X X
XXXXXXXXXXXXXXXXXXXX 8K
X X
X BANK 1 X
X X
XXXXXXXXXXXXXXXXXXXX 4K
X X
X BANK 0 X
X PROGRAM X
XXXXXXXXXXXXXXXXXXXX 0

```

THESE ANSWERS GIVEN BELOW WILL TEST THE CONFIGURATION IN THE
GIVEN EXAMPLE. ANSWERS:

TO TEST	-A- PORT	-B- PORT
	1) Y	1) N
	2) 1	4) 2
	3) 1	5) 1

PROGRAM CONVERSATION

MULTI DRIVE MODE? (YES-NO)

MULTI DISK MODE IS A MODE IN THE PROGRAM WHICH ALLOWS THE OPERATOR TO EXERCISE ALL THE DISKS ON THE SYSTEM WITHOUT RE-STARTING THE PROGRAM. THE PROGRAM, AFTER EXERCISING ONE DISK WILL REPORT A MESSAGE TELLING THE OPERATOR WHICH DISK WILL BE SELECTED NEXT, AND THEN THE PROGRAM WILL EXERCISE THAT DISK. WHEN A COMPLETE PASS IS ACCOMPLISHED, A PASS COMPLETE WILL BE REPORTED AND THE TEST WILL RECYCLE.

IF THE ANSWER TO THE MULTI DRIVE MODE WAS "NO", THE FOLLOWING QUESTION IS ASKED.

UNIT #

THE OPERATOR CAN NOW SELECT THE UNIT HE WISHES TO TEST BY TYPING THE UNIT NUMBER.

OPTIONAL WORD COUNT (YES-NO)

IF THE OPERATOR ANSWERS "NO" TO THIS QUESTION THE NEXT QUESTION WILL BE DELETED FROM THE CONVERSATION.

WD CT

THE OPERATOR CAN SPECIFY ANY LENGTH TRANSFER FROM 1(8) TO 60000(8) WORDS. THE NORMAL TRANSFER LENGTH IS N(8) WORDS WHERE N IS THE MAXIMUM BUFFER SIZE FOR THE AVAILABLE CORE. IN EITHER CASE, BUFFER WILL NOT EXCEED 24 K.

THIS PROGRAM MAPS THE SYSTEM IN 4K SEGMENTS. IF THERE IS A 1K BLOCK OF MEMORY ON THE SYSTEM THAT YOU WOULD LIKE TO REACH, YOU CAN TYPE IN THAT 4K BANK # AND THEN SPECIFY A WC OF 2000.

IF THE WORD COUNT NUMBER TYPED, IS LARGER THAN THE CORE SIZE GIVEN IN THE SETUP ROUTINE, THE QUESTION WILL BE REPEATED.

OPTIONAL DSK ADDR (YES-NO)

IF THE ANSWER TO THIS QUESTION IS NO, THE WHOLE DISK WILL BE WRITTEN AND THE NEXT QUESTION IS NOT ASKED.

DSK ADDR

THE OPERATOR CAN NOW SPECIFY THE STARTING SECTOR

PATTERN NO.?

THIS GIVES THE OPERATOR THE OPTION OF SELECTING ALL THE DATA PATTERNS (22) OR ANY ONE DATA PATTERN, SIMPLY BY TYPING THE DATA PATTERN NUMBER DESIRED.

PATTERN	0	=	000000
"	1	=	177777
"	2	=	031463
"	3	=	066666
"	4	=	100001
"	5	=	107070
"	6	=	070707
"	7	=	052525
"	10	=	125252
"	11	=	177737
"	12	=	146314
"	13	=	136363
"	14	=	063636
"	15	=	000001
"	16	=	100005
"	17	=	155555
"	20	=	133333
"	21	=	RANDOM DATA
"	22	=	RUN ALL DATA PATTERNS UNDER PROGRAM CONTROL

IN THIS SECTION OF THE PROGRAM PARAMETER CONVERSATION MODE, THE OPERATOR CAN SELECT ANY ONE OR ALL THREE OF THE CONTROL FUNCTIONS TO BE EXECUTED. THE NORMAL SEQUENCE OF DISK FUNCTIONS UNDER PROGRAM CONTROL ARE WRITE, WRITE CHECK, AND THEN READ. BY ENTERING THE CONVERSATION MODE THE OPERATOR HAS GAINED COMPLETE CONTROL OVER THE DISK FUNCTIONS. HE MUST SPECIFY YES OR NO TO ALL OF THE FOLLOWING QUESTIONS.

WRITE? (YES - NO)
READ? (YES - NO)
WRITE CHECK? (YES - NO)

TO PERFORM A WRITE CHECK ONLY, THE OPERATOR MUST FIRST WRITE SOME KNOWN DATA ON THE DISK. THIS COURSE OF ACTION ALSO PREVAILS FOR A READ ONLY OPERATION.
* IF AN ERROR OCCURS IN THE LINE THE OPERATOR IS TYPING, DEPRESS THE RUB-OUT KEY AND RETYPE ANSWER.
ALL ANSWERS SHOULD BE FOLLOWED BY A CARRIAGE-RETURN

5.3 ROUTINE ABSTRACTS

ADDRESS TEST

THIS TEST WRITES EACH SECTOR WITH ITS OWN ADDRESS THEN READS IT BACK AND COMPARES IT FOR THE CORRECT DATA.

RANEX - RANDOM DATA, ADDRESS AND WORD COUNT TEST

THIS ROUTINE TESTS THE ABILITY OF THE SYSTEM TO ACCESS RANDOM ADDRESSES WITH RANDOM DATA. ONE SECTOR OF RANDOM DATA IS WRITTEN AT A STARTING RANDOM ADDRESS ON THE DISK. IT IS THEN WRITE CHECKED AND READ. ALL ERRORS ARE REPORTED. THIS IS REPEATED 1000 TIMES.

DATA RELIABILITY - DATA PATTERN TEST

IN THIS PORTION OF THE TEST, THE RELIABILITY OF THE DISK SURFACE IS TESTED BY WRITE, WRITE CHECK, AND READ FUNCTIONS. THE ROUTINE FIRST WRITES THE COMPLETE SURFACE WITH A SET DATA PATTERN, THEN A WRITE CHECK OF THE COMPLETE SURFACE IS ACCOMPLISHED, THUS REPORTING ALL ERRORS BETWEEN THE DATA WRITTEN AND THE DATA IN MEMORY. THE DISK IS THEN READ. THE DATA READ FROM THE DISK IS COMPARED AGAINST THE KNOWN DATA PATTERN. THIS COMPARE IS TAKING PLACE THE SAME TIME THE DISK IS BEING READ. THE BUFFER IS CLEARED AS IT IS BEING COMPARED. IF THERE ARE DATA BUFFERS ON -A- AND -B- PORTS, THE DATA TEST WILL TRANSFER DATA OVER -A- PORT ON ODD PASSES AND OVER -B- PORT ON EVEN PASSES.

5.4 SUBROUTINE ABSTRACTS

5.4.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION "LAD". IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.4.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.4.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS;

```
ADR   CS1 = ----- CS2 = ----- ER = -----  
GOOD      = ----- BAD = -----
```

WHERE;

CS1,CS2,ER ETC.	= R811 DISK REGISTERS.
GOOD	= EXPECTED DATA.
BAD	= DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

IF SW0 IS SET, A DRIVE WILL BE DROPPED FROM THE TEST SEQUENCE AFTER 20 ERRORS. THE PROGRAM WILL STATE WHICH DRIVE WAS DROPPED AND ON WHICH PASS IT WAS DROPPED. IF ALL THE DRIVES HAVE BEEN DROPPED, THE PROGRAM WILL TYPE "TESTING UNIT 0" AND HALT, INDICATING THAT IT COULD NOT FIND ANY MORE DRIVES ON THE SYSTEM TO TEST.

7. RESTRICTIONS

THIS DIAGNOSTIC WILL TEST -B- PORT, ONLY IF THE CPU CAN ACCESS THAT MEMORY ON -B- PORT.

8. MISCELLANEOUS

8.1 EXECUTION TIME

PASS COMPLETE WILL BE TYPED OUT AT END OF PASS. IT WILL TAKE 10 TO 20 MINUTES TO COMPLETE A PASS DEPENDING ON THE TYPE OF DRIVE BEING TESTED AND THE SIZE OF THE SYSTEM.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

0.3 POWER FAIL

THE STARTING ADDRESS FOR THE WRITE POWER FAIL TEST IS 244. WHEN ASKED, ENTER UNIT 0. THE PROGRAM WILL TELL THE OPERATOR WHEN TO POWER DOWN. WHEN THE SYSTEM IS POWERED UP, ONLY ONE ERROR IS ALLOWED. THE

STARTING ADDRESS FOR THE WRITECHECK POWER FAIL TEST IS 250. HERE AS IN THE WRITE POWER FAIL TEST, THE PROGRAM WILL TELL THE OPERATOR WHEN TO POWER DOWN. WHEN THE POWER COMES BACK, NO ERRORS SHOULD OCCUR.

82		\$KMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS
627	TST1	ADDRESS TEST
705	TST2	DATA TEST
881	TST3	RANDOM ADDRESS RANDOM DATA TEST
1023	TST4	TEST FOR MULTI DISK MODE
1969		\$DONE - BELL AND SCOPE ROUTINE
2161		\$TYPE - TTY TYPEOUT ROUTINE
2201		\$SCOPE - SCOPE LOOP HANDLER
2240		\$HLT - HLT ROUTINE (ERROR TYPEOUT)
2270		\$OCTAL - OCTAL TYPEOUT ROUTINE
2318		\$POWER - POWER DOWN AND UP ROUTINES
2358		\$TYPEA - 18 BIT ADDRESS TYPER
2415		\$TRAP - TRAP HANDLER
2447		\$RDLIN - TTY INPUT ROUTINE
2508		\$RDOCT - OCTAL INPUT ROUTINE
2702		\$TYPED - CONVERT BINARY TO DECIMAL AND TYPE ROUTINE


```

1      .TITLE  MAINDEC-11-DERSC-E      RH11-R803-R803/LA-R804 DATA AND RELIABILITY TEST
2      ;COPYRIGHT 1973,1974,1975,1976 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
3      ;PROGRAM BY STANLEY HARACKIEWICZ
4
5      ;          SWITCH          USE
6      ;          -----          -----
7      100000      SW15= 100000      ;HALT ON ERROR
8      040000      SW14= 40000      ;LOOP ON FUNCTION
9      020000      SW13= 20000      ;INHIBIT ERROR TYPEOUTS
10     010000      SW12= 10000      ;INHIBIT COMPARISON
11     004000      SW11= 4000      ;HALT ON COMPLETION OF TRANSFER
12     002000      SW10= 2000      ;CONVERSATION MODE
13     001000      SW9= 1000      ;LOOP ON ERROR
14     000400      SW8= 400      ;DATA RELIABILITY TEST MODE
15     000200      SW7= 200      ;WAIT IN BACKGROUND TEST
16     000100      SW6= 100      ;OPTIONAL TYPEOUT OF RETRY ERRORS
17     000040      SW5= 40      ;INHIBIT PASS COUNT AND UNIT #
18     000020      SW4= 20      ;ALLOWS 8 LOCATIONS TO BE TESTED IN COMPARE ROUTINE
19     000010      SW3= 10      ;TYPE OUT TOTAL # OF ERRORS
20     000004      SW2= 4      ;INHIBIT MEMORY MANAGEMENT
21     000002      SW1= 2      ;DATA TEST ONLY
22     000001      SW0= 1      ;DROP DRIVE AFTER 20 ERRORS
23     000000      ;          ;TRAP CATCHER FROM 0 - 776
24     000046      ;          ;HOOKS FOR ACT 11
25     000046      ;          ;
26     000052      ;          ;
27     000052      ;          ;
28     000174      ;          ;
29     000174      ;          ;
30     000176      ;          ;
31     000176      ;          ;
32
33
34      ;NOTE:  FOR PROGRAM TO BE RESTARTED AT 200 IT MUST BE RELOADED
35      ;          INTO MEMORY DUE TO ONCE ONLY CODE.
36
37
38     000200      ;          ;
39     000200      000137      001230      ;          ;
40
41     000210      ;          ;
42     000210      012706      000500      ;          ;
43     000214      000137      003224      ;          ;
44
45     000220      ;          ;
46     000220      012706      000500      ;          ;
47     000224      000137      002374      ;          ;
48
49     000230      000137      015154      ;          ;
50
51     000234      000137      003316      ;          ;
52
53     000240      000137      005050      ;          ;
54     000244      000137      012466      ;          ;
55     000250      000137      013014      ;          ;

```

;SOFTWARE SWITCH REGISTER LOCATION

;START TEST

;SETUP STACK
;RESTART ADDR

;CONVERSATION MODE WITHOUT
;DATA BUFFER QUESTIONS

;RESTORE LOADER

;TRACK AND SECTOR SELECT TEST
;WRITE EACH WORD ADDR ON ITSELF AND READ IT BACK
;RANDOM ADDRESS, DATA TEST
;DISK WRITE POWER FAIL TEST
;DISK WRITE CHECK POWER FAIL TEST

```

56                                     ;RH11 DATA PATTERNS
57
58 000254 000000 PAT0: 0
59 000256 177777 PAT1: 177777
60 000260 031463 PAT2: 031463
61 000262 066666 PAT3: 066666
62 000264 100001 PAT4: 100001
63 000266 107070 PAT5: 107070
64 000270 070707 PAT6: 070707
65 000272 052525 PAT7: 052525
66 000274 125252 PAT10: 125252
67 000276 177737 PAT11: 177737
68 000300 146314 PAT12: 146314
69 000302 136363 PAT13: 136363
70 000304 063636 PAT14: 063636
71 000306 000001 PAT15: 000001
72 000310 100005 PAT16: 100005
73 000312 155555 PAT17: 155555
74 000314 133333 PAT20: 133333
75                                     ;PAT21 RANDOM DATA
76
77                                     ;CLEAR ALL REGISTERS
78 000316 012777 000040 000510 .CLRDI MOV 040,ORSC82 ;CLEAR ALL REG
79 000324 013777 001160 000502 MOV UNNUM,ORSC82 ;GET UNIT 0
80 000332 000002 RTI
81

```


	.SBTTL	SKMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS	
82			
83			
84	177572	SR0=177572	;ADDRESS OF MEM MGMT REGISTER SR0
85	177574	SR1=177574	; " " " " SR1
86	177576	SR2=177576	; " " " " SR2
87	172516	SR3=172516	;ADDRESS OF MEM MGMT REGISTER SR3
88			
89	172300	KIPDR0=172300	;ADDRESS OF KERNEL 'I' PAGE
90	172302	KIPDR1=172302	;DESCRIPTOR REGISTERS
91	172304	KIPDR2=172304	
92	172306	KIPDR3=172306	
93	172310	KIPDR4=172310	
94	172312	KIPDR5=172312	
95	172314	KIPDR6=172314	
96	172316	KIPDR7=172316	
97			
98	172320	KDPDR0=172320	;ADDRESSES OF KERNEL 'D' PAGE
99	172322	KDPDR1=172322	;DESCRIPTOR REGISTERS
100	172324	KDPDR2=172324	
101	172326	KDPDR3=172326	
102	172330	KDPDR4=172330	
103	172332	KDPDR5=172332	
104	172334	KDPDR6=172334	
105	172336	KDPDR7=172336	
106			
107	172340	KIPAR0=172340	;ADDRESSES OF KERNEL 'I' PAGE
108	172342	KIPAR1=172342	;ADDRESS REGISTERS
109	172344	KIPAR2=172344	
110	172346	KIPAR3=172346	
111	172350	KIPAR4=172350	
112	172352	KIPAR5=172352	
113	172354	KIPAR6=172354	
114	172356	KIPAR7=172356	
115			
116	172360	KDPAR0=172360	;ADDRESSES OF KERNEL 'D' PAGE
117	172362	KDPAR1=172362	;ADDRESS REGISTERS
118	172364	KDPAR2=172364	
119	172366	KDPAR3=172366	
120	172370	KDPAR4=172370	
121	172372	KDPAR5=172372	
122	172374	KDPAR6=172374	
123	172376	KDPAR7=172376	

124				
125	000001		N=	1
126	104000		HLT=	EMT
127	177776		PS=	177776
128	177776		PSW=	PS
129	000007		BELL=	7
130	000000		R0=	00
131	000001		R1=	01
132	000002		R2=	02
133	000003		R3=	03
134	000004		R4=	04
135	000005		R5=	05
136	000006		SP=	06
137	000007		PC=	07
138	000001		BIT0=	1
139	000002		BIT1=	2
140	000004		BIT2=	4
141	000010		BIT3=	10
142	000020		BIT4=	20
143	000040		BIT5=	40
144	000100		BIT6=	100
145	000200		BIT7=	200
146	000400		BIT8=	400
147	001000		BIT9=	1000
148	002000		BIT10=	2000
149	004000		BIT11=	4000
150	010000		BIT12=	10000
151	020000		BIT13=	20000
152	040000		BIT14=	40000
153	100000		BIT15=	100000
154	000001	GOOD=	R1	
155	000000	BAD=	R0	

```

;INITALIZE FOR NEWTST
;SET HLT TO EMT FOR ERROR TYPEOUTS
;PROCESSOR STATUS
;PROCESSOR STATUS WORD
;BELL
;R0 - DEFINE REGISTERS
;R1
;R2
;R3
;R4
;R5
;R6 - STACK POINTER
;R7 - PROGRAM COUNTER
;BIT EQUATES

```

```

;FOR GOOD DATA
;FOR BAD DATA

```


156		000510			.#	510	
157	000510	005015	047125	052111	LOADSW:	.ASCIZ	<15><12>"UNIT 0 "
158	000516	021440	000040				
159	000522	005015	040506	040524	UNRECO:	.ASCIZ	<15><12>"FATAL"<15><12><12>
160	000530	006514	005012	000			
161							
162							
163	000535	015	047412	052120	.OPDR:	.ASCIZ	<15><12>"OPT DSK ADDR"
164	000542	042040	045523	040440			
165	000550	042104	000122				
166	000554	005015	040504	040524	DATA:	.ASCIZ	<15><12>"DATA "
167	000562	000040					
168							
169	000564	051127	020124	051105	WRTERR:	.ASCIZ	"WRT ERR"
170	000572	000122					
171							
172	000574	051127	020124	045503	WCKERR:	.ASCIZ	"WRT CK ERR"
173	000602	042440	051122	000			
174							
175	000607	122	020104	051105	RDERR:	.ASCIZ	"RD ERR"
176	000614	000122					
177							
178	000616	005015	042522	020103	RECOV:	.ASCIZ	<15><12>"REC RETRY CT "
179	000624	042522	051124	020131			
180	000632	052103	000040				
181							
182	000636	005015	000		CRLFLP:	.ASCIZ	<15><12>
183							
184	000641	040	054450	047440	YORN:	.ASCIZ	" (Y OR N)"
185	000646	020122	024516	000			
186							
187	000653	015	030412	052123	STABUF:	.ASCIZ	<15><12>"1ST 4K BANK 0 "
188	000660	032040	020113	040502			
189	000666	045516	021440	000040			
190							
191	000674	005015	020043	043117	BUFSIZ:	.ASCIZ	<15><12>"# OF 4K BANKS?"
192	000702	032040	020113	040502			
193	000710	045516	037523	000040			
194							
195	000716	005015	040522	042116	RANDM:	.ASCIZ	<15><12>"RANDOM "
196	000724	046517	000040				
197	000730	005015	042524	052123	TSTNG:	.ASCIZ	<15><12>"TESTING "
198	000736	047111	020107	000			
199	000743	015	050012	053517	PDOWN:	.ASCIZ	<15><12>"POWER DOWN"
200	000750	051105	042040	053517			
201	000756	000116					
202	000760	005015	040520	044522	PARERR:	.ASCIZ	<15><12>"PARITY ERR"
203	000766	054524	042440	051122			
204	000774	000					
205							
206		000776			.EVEN		
207							

208	001000	. =	1000	
209				
210	001000	000000	ICNT:	0
211	001002	000000	ERRORS:	0
212	001004	000000	PCNT:	0.0
213	001010	000000	LAD:	0
214	001012	000000	HLTADR:	0
215	001014	001000	FILCHR:	1000
216	001016	177564	TPB:	177564
217	001020	177560	TKB:	177560
218	001022	177562	TKB:	177562
219	001024	177566	TPB:	177566
220	001026	177570	SWR:	177570
221	001030	177570	DISPLAY:	177570
222				
223				
224				
225	001032	172040	RSCB1:	172040
226	001034	172050	RSCB2:	172050
227	001036	172042	RSWC:	172042
228	001040	172044	RSBA:	172044
229	001042	172046	RSDA:	172046
230	001044	172052	RSDS:	172052
231	001046	172054	RSER:	172054
232	001050	172056	RSAS:	172056
233	001052	172060	RLA:	172060
234	001054	172062	RSDB:	172062
235	001056	172064	RSNR:	172064
236	001060	172066	RSDT:	172066
237	001062	000204	RSVEC:	204
238				
239				
240				
241	000002		DB=2	
242	000004		DA=4	
243	000010		WC=10	
244	000020		BA=20	
245	000040		DS=40	
246	000100		AS=100	
247	000204		LA=204	
248	000220		MR=220	
249	000240		DT=240	
250				
251	001064	000206	STATUS:	206
252	001066	000200	PRIORITY:	BIT7

;LH = ITERATION COUNT ;RH = TEST NO.
 ;ERROR COUNT
 ;2 WORD PASS COUNT
 ;LOOP ADDRESS FOR SCOPE
 ;ADDRESS OF LAST HLT INSTRUCTION EXECUTED
 ;FILCHR=0 (CHAR) ;FILCHR+1=2 (COUNT)
 ;OUTPUT STATUS REGISTER

 ;OUTPUT BUFFER
 ;SWITCH REGISTER
 ;DISPLAY REGISTER

 ;DISK I/O REGISTERS

 ;DISK CONTROL + STATUS REGISTER
 ;DISK CONTROL + STATUS REGISTER
 ;WORD COUNT REGISTER
 ;BUS ADDRESS
 ;DISK ADDRESS (DESIRED ADDRESS)
 ;DRIVE STATUS
 ;ERROR REG.
 ;ATTENTION SUMMARY
 ;LOOK AHEAD
 ;DATA BUFFER REGISTER
 ;MAINTENANCE REGISTER
 ;DRIVE TYPE REGISTER
 ;INTERUPT RSVEC

 ;BIT ASSIGNMENTS FOR ERROR TYPE OUTS

 ;DATA BUFFER
 ;DESIRED ADD
 ;WORD COUNT
 ;BUS ADDRESS
 ;DRIVE STATUS
 ;ATTENTION SUMMARY
 ;LOOK AHEAD
 ;MAINTENANCE
 ;DRIVE TYPE

 ;DISK INTERRUPT STATUS
 ;DISK PRIORITY LEVEL

253		000000	RW=6	;R/W IN PDR REG
254		000000	UP=0	;UP BITY IN PDR REG
255		000250	MNVEC=250	;ADDR OF MEM MGMT ERROR TRAP
256	001070	000000	STANEM: 0	;STARTING LOC FOR -A- PORT
257	001072	000000	SAVAST: 0	;SAVE LOC FOR STANEM
258	001074	000000	STBCOM: 0	;STARTING LOC FOR -B- PORT
259	001076	000000	SAVCPU: 0	;SAVE LOC FOR CPUBM
260	001100	000000	SAVMA: 0	;STARTING ADDR FOR -A- PORT WITH MEM MGMT
261	001102	000000	SAVMB: 0	;STARTING ADDR FOR B PORT W/MEM MGMT
262	001104	000000	SAVMGC: 0	;STARTING LOC FOR CPU W/MEM MGMT
263	001106	000000	SIZEAP: 0	;SIZE OF A PORT
264	001110	000000	SIZEBP: 0	;SIZE OF B PORT
265	001112	000000	WDCTB: 0	;WC FOR A PORT
266	001114	000000	A0B1: 0	;FLAG FOR PORT BEING TESTED
267	001116	000000	VADDR: 0	;VIRTUAL ADDR
268	001120	000000	PHADDR: 0	;PHYSICAL ADDR
269	001122	000000	FLAG2: 0	;FLAG FOR RESTART AND FOUND DRIVE
270	001124	000000	DROP: 0	;BAD UNITS ON SYSTEM THAT GET DUMPED

;DISCRIPTION OF FLAG2

;BIT0 = RESTART
 ;BIT1 = FOUND DRIVE
 ;BIT2 = ERROR DO A CRLF FOR UNIT 0
 ;BIT3 = DOING COMPARE
 ;BIT4 = SET A16 IN CS1
 ;BIT5 = SET A17 IN CS1
 ;BIT6 = SET IF MEMORY HAS ALREADY BEEN FOUND
 ;BIT7 = WHEN SET MAKE WC UP TO 20K
 ;BIT8 = FOUND MEMORY ON -B- PORT
 ;BIT9 = POWER DID FAIL
 ;BIT10 = WAITING IN BACKGROUND TEST
 ;BIT11 = PARITY ERROR ROUTINE
 ;BIT12 = POWER FAIL TEST
 ;BIT13 = IN MAP ROUTINE
 ;BIT14 = IN POWER FAIL OR CONVEPATION MODE
 ;BIT15 = ERROR IN POWER FAIL

;DISCRIPTION OF FLAG

;BIT0 = USED FOR WRITE COUNTER
 ;BIT1 = USED FOR WRITE COUNTER
 ;BIT2 = TRANSFER MODE 64K
 ;BIT3 = OPTIONAL DMA
 ;BIT4 = TEST -B- PORT
 ;BIT5 = LAST DISK BUFFER FLAG
 ;BIT6 = PROGRAM IS IN ADDRESS OR RANDCM TEST
 ;BIT7 = ERROR DURING TRANSFER
 ;BIT8 = DATA TEST ONLY
 ;BIT9 = MULTIPORT
 ;BIT10 = READ
 ;BIT11 = WRITE CHECK
 ;BIT12 = WRITE
 ;BIT13 = PROGRAM CONTROL MODE

307					
308					
309	001126	000000	FLAG:	0	;TEST REGISTER
310	001130	000000	WRDCT:	0	;WORKING WORD COUNT
311	001132	000000	TRACK:	0	;WORKING DAE
312	001134	000000	DMA:	0	;WORKING DAR
313	001136	000000	PATNU:	0	;DATA PATTERN INDEX
314	001140	000000	BUF:	0	;WORKING DATA BUFFER (OUT-IN)
315	001142	000000	TDMA:	0	;TEMP DAR
316	001144	000000	SWRDCT:	0	;STANDARD WORD COUNT
317	001146	000000	ERCOUNT:	0	;ERROR COUNT FOR MESSAGES.
318	001150	000000	SAVE:	0	
319	001152	000000	HRDER:	0	;POINTER FOR HARD ERROR
320	001154	000000	BLOCK:	0	
321	001156	000000	PASSC:	0	
322	001160	000000	UNNUM:	0	;UNIT CURRENTLY BEING TESTED
323	001162	000000	UNITSV:	0	;SET BIT=UNIT ON BUS
324	001164	000000	UNCMP:	0	;FOR COMPARING FOR 0 OF DEVICE
325	001166	000000	RS04DT:	0	;FLAG FOR RS04
326	001170	000000	NUMS:	0	;WORK LOC FOR NUMBER INPUTS
327	001172	000000	CMD:	0	;LOC FOR CS2 COMMANDS
328	001174	000000	SWITCH:	0	;FLAG FOR WHICH RANDOM NUMBER GEN
329	001176	000000	INTPLG:	0	;FLAG FOR INTERRUPT
330	001200	000000	LOPCNT:	0	;ERROR FLAG AND LOOP COUNTER FLAG
331	001202	000000	WRITER:	0	;CONTAINS 0 OF WRITE ERRORS
332	001204	000000	WCERR:	0	;CONTAINS 0 OF WRITE CHECK ERRORS
333	001206	000000	READER:	0	;CONTAINS 0 OF READ ERRORS
334	001210	000000	COMERR:	0	;CONTAINS 0 OF COMPARE ERRORS
335	001212	000000	MMAVA:	0	;MEM MGMT AVAILABLE INDICATOR
336	001214	000000	SAVWC:	0	;SAVE LOC FOR CONVERSATION WC ROUTINE
337	001216	000000	FLAG3:	0	;LOOP IN ADDRESS + RANDOM TST FLAG
338	001220	000000	SAVWC0:	0	;SAVE WC SIZE FOR -B- PORT
339					
340					
341					
342	001222	000000			
343	001224	000000			
344	001226	000000			

;RH11 DEDICATE REGISTERS (MEMORY)

;RH11 WORK REGISTERS
(CAN BE CHANGED IN ANY ROUTINE)

WORK: 0
WORK1: 0
WORK2: 0

345	001230	005037	020116		SETSWI: CLR	SWI	
346	001234	012706	000500		BEGIN: NOV	0500,SP	;SET STACK TO *** 500 ***
347	001240	012737	016254	000024	NOV	0,POWER,0024	;SET UP PF VECTOR
348	001246	012737	000340	000026	NOV	0340,0026	;LOCK OUT THE WORLD
349	001254	012737	015716	000030	NOV	0,HLT,0030	;SET ENT VECTOR
350	001262	012737	000340	000032	NOV	0340,0032	;LOCK UP
351	001270	012737	016622	000034	NOV	0,TRAP,0034	;SET TRAP VECTOR
352	001276	012737	000340	000036	NOV	0340,0036	;LOCK UP
353	001304	005037	001000		CLR	ICNT	;INIT ICNT
354	001310	005037	001010		CLR	LAD	;INIT LAD
355	001314	032737	000001	020116	BIT	0BIT0,SWI	
356	001322	001001			BNE	20	
357	001324	104444			SUSWR		;SIZE FOR SWITCHLESS
358	001326	042737	177677	001126	BIC	0177677,FLAG	;CLEAR FLAG
359	001334	042737	177776	001122	BIC	0177776,FLAG2	;CLEAR ALL EXECPT RESTART
360	001342	005037	001216		CLR	FLAG3	;CLEAR LOOP IN ADDRESS + RANDOM TST FLAG
361	001346	032737	000001	001122	BIT	0BIT0,FLAG2	;IS THIS THE FIRST TIME?
362	001354	001002			BNE	10	;NO
363	001356	004737	020120		JSR	PC,LDR	;SAVE LOADER
364	001362	000005		101	RESET		;CLEAR THE WORLD
365	001364	012737	000340	177776	NOV	0340,PS	;LOCK UP INTERRUPT LEVELS
366	001372	004537	012432		JSR	RS,ERRCL	;CLEAR ERROR COUNTER + PASS CNT
367	001376	005037	001212		CLR	NHAVA	;CLEAR MEM NGMT FLAG
368	001402	005037	001114		CLR	A0B1	;TEST A PORT FIRST
369	001406	032777	000004	177412	BIT	0BIT2,0SWR	;WANT MEM NGMT?
370	001414	001021			BNE	30	;NO
371	001416	012737	001444	000004	NOV	050,4	;SET TIMEOUT TRAP
372	001424	012737	000340	000006	NOV	0340,6	;SET PS
373	001432	005037	177572		CLR	00SR0	;IS MEM NGMT AVAILABLE?
374	001436	005137	001212		COM	NHAVA	;YES
375	001442	000401			BR	40	;CONT
376	001444	022626		501	CMP	(6)+,(6)+	;CLEAR STACK
377	001446	012737	000006	000004	401	NOV	06,4
378	001454	005037	000006		CLR	6	;TRAP
379	001460	032737	000001	001122	301	BIT	0BIT0,FLAG2
380	001466	001002			BNE	CALN	;IS THIS THE FIRST TIME
381	001470	000137	020210		JMP	SIZZAP	;NO
382	001474	004737	011472		JSR	PC,00EXTMEN	;SIZE BUFFERS
383	001500	004737	015202		JSR	PC,,HANK	;SET UP DATA BUFFERS
384	001504	032737	000001	001122	BIT	0BIT0,FLAG2	;TURN ON PARITY MEM
385	001512	001006			BNE	30	;1ST TIME ?
386	001514	013737	001144	001214	NOV	SWRDCT,SAVNC	;NO
387	001522	013737	001112	001220	NOV	WDCTB,SAVNCB	;SAVE WC FOR CONVERSATION MODE COMPARE
388	001530	052737	000001	001122	301	BIS	0BIT0,FLAG2
389	001536	005037	001134		CLR	DMA	;SET 1ST TIME FLAG
390	001542	005037	001136		CLR	PATNU	;CLEAR DAR REGISTERS
391	001546	013737	001144	001130	NOV	SWRDCT,WRDCT	;CLEAR PATTEN COUNT
392	001554	032777	000002	177244	BIT	0BIT1,0SWR	
393	001562	001403			BEQ	20	;DATA TEST ONLY?
394	001564	052737	002000	001126	BIS	0BIT10,FLAG	;NO
395	001572	032777	002000	177226	201	BIT	0BIT10,0SWR
396	001600	001007			BNE	10	;YES
397	001602	052737	074000	001126	BIS	074000,FLAG	;ENTER CONVERSATION MODE?
398	001610	004537	010214		JSR	RS,RESTOR	;YES GO TO CONVERSATION MODE
399	001614	000137	003224		JMP	ADTST	
400	001620	000137	002220	101	JMP	00CONN	;RESTORE ORIGINAL WD CNT
							;ENTER CONVERSATION MODE


```

401          ;FIND OUT HOW MANY DRIVES
402          ;FIRST TEST RSAS
403
404 001624 012701 000010          DRVENO: MOV      00.,R1          ;PUT 0 INTO R1 FOR COUNT
405 001630 042737 000002 001122          BIC      0BIT1,FLAG2      ;CLEAR FOUND DRIVE FLG
406 001636 012777 000000 177170          MOV      00,0RSCS2      ;SET DEVICE TO ZERO
407 001644 012777 000007 177174          TRY:  MOV      07,0RSEB      ;CAUSE AN ERROR +SETS BIT IN AS REG
408 001652 005301          DEC      R1          ;DO A MAXIMUM OF 16 TIMES
409 001654 001403          BEQ      DVNUM          ;TESTED FOR ALL DRIVES GET OUT
410 001656 005277 177152          INC      0RSCS2      ;INCREMENT DRIVE UNIT
411 001662 000770          BR       TRY          ;REPEAT FOR NEXT DRIVE
412 001664 017737 177160 001162          DVNUM: MOV      0RSAS,UNITSV      ;SAVE
413 001672 043737 001124 001162          BIC      DROP,UNITSV      ;DROP BAD DRIVES
414 001700 012737 000401 001164          MOV      0401,UNCMP      ;SETUP TO CMP WITH UNITSV
415 001706 012737 000000 001160          MOV      00,UNNUM          ;PUT 0 INTO UNIT NO.
416 001714 032777 000040 177104          BIT      0BITS,0SWR      ;INHIBIT TYPE OUT?
417 001722 001005          BNE      STTEST          ;YES
418 001724 104402 000730          TYPE      ,TSTNG
419 001730 042737 000004 001122          BIC      0BIT2,FLAG2      ;CLEAR ERROR FLAG
420 001736 033737 001164 001162          STTEST: BIT      UNCMP,UNITSV      ;IS THIS DRIVE ON THE SYSTEM
421 001744 001473          BEQ      TRYNX          ;NO
422 001746 013777 001160 177060          UNTP:  MOV      UNNUM,0RSCS2      ;YES PUT UNIT 0 INTO CS2
423 001754 005037 001166          CLR      RS04DT          ;CLEAR DRIVE TYPE FLAG
424 001760 022777 000004 177072          CMP      04,0RSDT          ;RS03LA?
425 001766 001004          BNE      00          ;NO
426 001770 012737 000004 001166          MOV      04,RS04DT          ;SET DRIVE TYPE FLAG
427 001776 000422          BR       10          ;CONT
428 002000 005777 177054          TST      0RSDT          ;IS THIS A RS03?
429 002004 001417          BEQ      10          ;YES
430 002006 022777 000001 177044          CMP      01,0RSDT          ;IS THIS A RS03 4US?
431 002014 001413          BEQ      10          ;YES
432 002016 022777 000002 177034          CMP      02,0RSDT          ;IS THIS A RS04?
433 002024 001404          BEQ      60          ;YES
434 002026 022777 000003 177024          CMP      03,0RSDT          ;RS04?
435 002034 001037          BNE      TRYNX          ;GET A NEW NUMBER
436 002036 052737 177777 001166          DIS      0-1,RS04DT          ;YES RS04
437 002044 032737 040000 001122          BIT      0BIT14,FLAG2      ;IN POWER FAIL OR CONVERSATION?
438 002052 001401          BEQ      70          ;NO
439 002054 000207          RTS      PC          ;YES
440 002056 032777 000200 176760          BIT      0BIT7,0RSDS      ;IS THIS DRIVE READY ?
441 002064 001423          BEQ      TRYNX          ;NO GET ANOTHER DRIVE
442 002066 032777 000040 176732          BIT      0BITS,0SWR      ;TYPEOUT?
443 002074 001016          BNE      40          ;NO
444 002076 032737 000004 001122          BIT      0BIT2,FLAG2      ;WAS THERE AN ERROR?
445 002104 001402          BEQ      50          ;NO
446 002106 104402 000636          TYPE      ,CRLF
447 002112          ;
448 002112 013746 001160          MOV      UNNUM,-(6)          ;PUT UNNUM ON STACK
449 002116 104406          TYPES          ;TYPE STACK IN OCTAL - SUPRESS
450 002120 104402 000040          TYPE      ,40          ;TYPE SPACE
451 002124 042737 000004 001122          BIC      0BIT2,FLAG2      ;CLEAR ERROR FLAG
452 002132 000426          BR       NONGO          ;NOW TEST

```



```

453 002134 006337 001164      TRYNX:  ASL      UNCHP      ;CHECK NEXT BIT FOR DRIVE
454 002140 103403              BCS      CHCKDV     ;DID WE TEST ANY REG?
455 002142 005237 001160              INC      UNNUM     ;INC UNIT 0
456 002146 000673              BR       STTEST     ;CHECK FOR NEXT DRIVE
457
458                                ;THIS PROGRAM WILL DEFAULT TO TESTING UNIT 0 IF IT CAN NOT FIND ANY DRIVES
459 002150 032737 000002 001122  CHCKDV:  BIT      @BIT1,FLAG2 ;FOUND DRIVE?
460 002156 001012              BNE      DONEE      ;YES WE DID TEST A DRIVE
461 002160 012737 100000 001164      MOV      @100000,UNCHP ;NO DRIVES TESTED, COULD NOT SET
462 002166 005037 001160              CLR      UNNUM     ;ANY AS BITS, THUS DEFAULTS TO 0
463 002172 013746 001160              MOV      UNNUM,-(6) ;PUT UNNUM ON STACK
464 002176 104406              TYPES     ;TYPE STACK IN OCTAL - SUPPRESS
465 002200 000000              HALT      ;COULD NOT SET ANY ATA BITS
466
467                                ;BY SETTING ERROR BITS
468 002202 000402              BR       NONGO      ;GO BACK AND USE OTHER DIAG.
469 002204 000137 013446 001122  DONEE:  JMP      OUT      ;DEFAULT TO DRIVE 0
470 002210 052737 000002 001122  NONGO:  BIS      @BIT1,FLAG2 ;GET OUT
471 002216 000207              RTS      PC      ;FOUND DRIVE
472
473                                ;ENTER OPERATOR CONVERSATION MODE
474
475                                CONN:
476 002220 104402 007224              TYPE     ,,+2      ;,ASCIZ <15><12>"-A- PORT"
477 002240 004737 003300              JSR      PC,CHPY   ;COMPARE FOR YES
478 002244 001405              BEQ      20      ;YES
479 002246 012737 177777 001114      MOV      @-1,A0B1   ;B PORT
480 002254 000137 002320              JMP      10      ;TEST -B- PORT
481 002260 104402 000653 20:        TYPE     ,STABUF
482 002264 104420              RDOCT
483 002266 012637 001070              NOV      (6)+,STAMEN ;START BUFFER AT 4K
484 002272 104402 000674 30:        TYPE     ,BUFSIZ
485 002276 104420              RDOCT
486 002300 012637 001106              NOV      (6)+,SIZEAP ;GET NUMBER
487 002304 022737 000006 001106      CMP      @6,SIZEAP ;SAVE IT
488 002312 002767              BLT      30      ;CAN ONLY XFER 24K
489 002314 000137 002362              JNP      NOPORT   ;GREATER THEN 24K
;GET OUT

```

490	002320	104402	000653	18:	TYPE	,STABUF	
491	002324	104420			RDOCT		;GET ANS
492	002326	012637	001074		MOV	(6)+,STBCOM	;AND SAVE IT
493	002332	104402	000674	48:	TYPE	,BUFSIZ	
494	002336	104420			RDOCT		;GET ANS
495	002340	012637	001110		MOV	(6)+,SIZEBP	;SAVE IT
496	002344	022737	000006	001110	CMP	#6,SIZEBP	;GREATER THEN 24K?
497	002352	002767			BLT	48	;YES ASK AGAIN
498	002354	052737	000100	001126	BIS	#BIT6,FLAG	;SET B PORT FLAG
499	002362	004737	011472		JSR	PC,EXTMEM	;CAL BUFFERS AND WC
500	002366	013737	001144	001130	MOV	SWRDC,WRDCT	;GET STANDARD WC
501	002374	052737	002000	001126	A1:	BIS	#BIT10,FLAG
502	002402	004537	012432		JSR	R5,ERRCL	;SET BIT FOR DATA TEST ONLY
503	002406	042737	174040	001126	BIC	#174040,FLAG	;CLEAR ERROR CNT + PASS CNT
504	002414	104402	002420		TYPE	,,+2	;CLEAR MULTI FLAG MODE +PATTERN SELECT
505	002436	004737	003300		JSR	PC,CNRY	;ASCIZ <15><12>"MULTI DRIVE"
506	002442	001004			BNE	DATTES	;COMPARE FOR YES
507	002444	052737	004000	001126	BIS	#BIT11,FLAG	;ANS IS NO
508	002452	000434			BR	ASKWC	;SET BIT FOR MULTI DRIVE
509	002454	104402	000510		DATTES:	TYPE	
510	002460	104420			RDOCT		
511	002462	012637	001170		MOV	(6)+,NUM0	;GET NUMBER
512	002466	022737	000010	001170	CMP	#10,NUM0	;CORRECT ?
513	002474	103767			BLO	DATTES	;NO
514	002476	013737	001170	001160	MOV	NUM0,UNNUM	;SET UNIT :
515	002504	004737	006540		JSR	PC,FNDTYP	;TEST FOR RS04 OR 03
516	002510	005002		18:	CLR	R2	;CLEAR WORK AREA
517	002512	000261			SEC		;SET CARRY
518	002514	006102		28:	ROL	R2	;SET BIT IN WORK
519	002516	005737	001170		TST	NUM0	;IS THIS THE RIGHT BIT FOR THE RIGHT DISK
520	002522	001403			BEQ	30	;YES
521	002524	005337	001170		DEC	NUM0	;NO TRY AGAIN
522	002530	000771			BR	28	;TEST AGAIN
523	002532	010237	001162	38:	MOV	R2,UNITSV	;SET DRIVE BIT IN UNITSV
524	002536	052737	000002	001122	BIS	#BIT1,FLAG2	;SET FOUND DRIVE FLAG
525							
526	002544				ASKWC:		
527	002544	104402	002550		TYPE	,,+2	;ASCIZ <15><12>"OPT WD CT"
528	002564	004737	003300		JSR	PC,CNRY	;COMPARE FOR YES
529	002570	001403			BEQ	WCCON	;YES
530	002572	004537	010214		JSR	R5,RESTOR	;RESTORE ORIGINAL WD CNT
531	002576	000444			BR	OPDAR	;CONT

532	002600				WCCON:			
533	002600	104402	002604		TYPE	..+2		:.ASCIZ <15><12>"WD CT "
534	002616	104420			RDOCT			
535	002620	012637	001170		MOV	(6)+,NUM8		:GET NUMBER
536	002624	005737	001170		TST	NUM8		:IS IT 0?
537	002630	001763			BEQ	WCCON		:YES ASK AGAIN FOR LENGTH
538	002632	013702	001214		MOV	SAVWC,R2		:GET STANDARD WC FOR -A- PORT
539	002636	005737	001114		TST	A0B1		: -A- PORT?
540	002642	001402			BEQ	18		:YES
541	002644	013702	001220		MOV	SAVWC,R2		:GET WC FOR -B- PORT
542	002650	005202			INC	R2		
543	002652	020237	001170		CMP	R2,NUM8		:IS NUM8 LESS THAN SWRDCT
544	002656	101750			BLOS	WCCON		:YES ASK FOR COUNT AGAIN
545	002660	013737	001170	001144	MOV	NUM8,SWRDCT		:OPERATING WORD COUNT
546	002666	013737	001144	001130	MOV	SWRDCT,WRDCT		
547	002674	005737	001114		TST	A0B1		:B PORT?
548	002700	001403			BEQ	OPDAR		:NO
549	002702	013737	001130	001112	MOV	WRDCT,WDCTB		:YES GET WC
550								
551	002710	104402	000535		OPDAR:	TYPE	..OPDR	
552	002714	004737	003300		JSR	PC,CMPY		:COMPARE FOR YES
553	002720	001037			BNE	OPPAT		:ANS IS NO
554	002722	052737	000040	001126	BIS	0BITS,FLAG		:SET OPTIONAL DMA FLAG
555	002730	104402	002734		TYPE	..+2		:.ASCIZ "0"
556	002736	104420			RDOCT			
557	002740	012637	001170		MOV	(6)+,NUM8		:GET NUMBER
558	002744	022737	000004	001166	CMP	84,R804DT		:R803LA?
559	002752	001004			BNE	38		:NO
560	002754	022737	003777	001170	CMP	03777,NUM8		:IS ADDR. CORRECT?
561	002762	000412			BR	28		
562	002764	005737	001166		TST	R804DT		:IS THIS A R804?
563	002770	001404			BEQ	18		:NO
564	002772	022737	017777	001170	CMP	017777,NUM8		:IS ADD. CORRECT?
565	003000	000403			BR	28		:GET OUT
566	003002	022737	007777	001170	CMP	07777,NUM8		:IS ADD. CORRECT?
567	003010	101737			BLOS	OPDAR		:NO
568	003012	013737	001170	001142	MOV	NUM8,TDNA		:TEMP SECTOR REGISTER

569	003020			OPPAT:			
570	003020	104402	003024		TYPE	..+2	;ASCIZ <15><12>"PATN 0"
571	003036	104420			RDOCT		
572	003040	012637	001170		MOV	(6)+,NUM8	;GET NUMBER
573	003044	022737	000023 001170		CMP	#23,NUM8	;TEST FOR CORRECT NO
574	003052	101762			BLOS	OPPAT	;ASK AGAIN
575	003054	005037	001136		CLR	PATNU	;CLEAR PATTERN 0
576	003060	022737	000022 001170		CMP	#22,NUM8	
577	003066	001411			BEQ	OPWRT	;DATA PATTERN UNDER PROGRAM CONTROL
578	003070	052737	100000 001126		BIS	#BIT15,FLAG	;SET PROGRAM FLAG
579	003076	013737	001170 001136		MOV	NUM8,PATNU	;OPERATOR WANTS TO SELECT DATA
580	003104	000241			CLC		
581	003106	006137	001136		ROL	PATNU	
582							
583	003112			OPWRT:			
584	003112	104402	003116		TYPE	..+2	;ASCIZ <15><12>"WRITE"
585	003126	004737	003300		JSR	PC,CMPY	;COMPARE FOR YES
586	003132	001003			BNE	OPRD	;ASK ABOUT WRITE CHECK
587	003134	052737	040000 001126		BIS	#BIT14,FLAG	;YES SET FLAG BIT
588							
589	003142			OPRD:			
590	003142	104402	003146		TYPE	..+2	;ASCIZ <15><12>"READ"
591	003156	004737	003300		JSR	PC,CMPY	;COMPARE FOR YES
592	003162	001003			BNE	OPWCK	
593	003164	052737	010000 001126		BIS	#BIT12,FLAG	;SET FLAG TO READ
594							
595	003172			OPWCK:			
596	003172	104402	003176		TYPE	..+2	;ASCIZ <15><12>"WRT CK"
597	003210	004737	003300		JSR	PC,CMPY	;COMPARE FOR YES
598	003214	001003			BNE	ADTST	
599	003216	052737	020000 001126		BIS	#BIT13,FLAG	
600							
601	003224	032737	004000 001126	ADTST:	BIT	#BIT11,FLAG	;ARE WE IN MULTI DRIVE MODE
602	003232	001402			BEQ	EXMFLG	;BRANCH IF NO.
603	003234	004737	001624		JSR	PC,DRVENO	;GET DRIVES TO BE TESTED
604	003240	042737	000004 001126	EXMFLG:	BIC	#BIT2,FLAG	;CLEAR XFER MODE FLAG
605	003246	032777	000400 175552		BIT	#BIT0,08WR	;XFER MODE?
606	003254	001403			BEQ	10	;NO
607	003256	052737	000004 001126		BIS	#BIT2,FLAG	;SET XFER MODE FLAG
608	003264	032737	002000 001126	10:	BIT	#BIT10,FLAG	;TEST FOR DATA TEST ONLY
609	003272	001414			BEQ	ADT1	;DO COMPLETE TEST
610	003274	000137	003710		JNP	DATAT	;DO DATA TEST ONLY
611							
612	003300	104402	000641	CMPY:	TYPE	,YORN	
613	003304	104422			RDLIN		
614	003306	122737	000131 017032		CMPB	#'Y,INPUT	;TEST FOR YES
615	003314	000207			RTS	PC	
616	003316	052737	100000 001216	ADTL:	BIS	#BIT15,FLAG3	;SET LOOP IN ADDRESS TEST FLAG GOT HERE
617							;BECAUSE PROGRAM WAS STARTED AT 234


```

618 ;RH11 ADDRESS TEST 01 (TRACK AND SECTOR SELECTION TEST)
619 ;WRITE 100(OCTAL) RS03, 200(OCTAL) RS04, WORDS IN EACH SECTOR
620 ;THE WORD CONTAINS THE ADDRESS OF EACH SECTOR
621 ;WHEN THE COMPLETE DISK IS WRITTEN READ
622 ;BACK EACH SECTOR AND COMPARE FOR THE CORRECT
623 ;DATA IN THE SECTOR
624 ;PS IS AT LEVEL 7 SO NO INTERRUPTS
625
626 003324 ADT1: ;ADDRESS TEST
627 ;*****
628 ;TEST 1 ADDRESS TEST
629 ;*****
630 003324 104400 TST1: SCOPE
631 003326 032737 000004 001126 ADT1A: BIT 0BIT2,FLAG ;XFER MODE?
632 003334 001402 BEQ 30 ;NO
633 003336 000137 003710 JMP DATAT ;YES
634 003342 012737 000340 177776 30: MOV 0340,PS ;LOCK UP PS
635 003350 012737 020000 017416 MOV 020000,OUTBUF ;START BUF AT 20000
636 003356 052737 000400 001126 BIS 0BIT0,FLAG ;SET TEST FLAG
637 003364 013737 001144 001150 MOV SWRDCT,SAVE ;SAVE STD WD COUNT
638 003372 005037 001134 CLR DMA ;CLEAR DISK ADD
639 003376 104426 CLRDV ;INIT DRIVE
640 003400 004737 006562 JSR PC,WHTHU ;GET WORD COUNT
641 003404 013737 001130 001144 50: MOV WRDCT,SWRDCT
642 003412 013737 017416 001140 20: MOV OUTBUF,BUF ;SET UP CURRENT ADDRESS
643 003420 104414 SEABUF: ERCLR ;CLEAR RS REGISTERS IF ERROR
644 003422 013700 017416 MOV OUTBUF,R0 ;SET UP ADDRESS BUFFER
645 003426 013701 001130 MOV WRDCT,R1
646 003432 013720 001134 XSEABUF: MOV DMA,(0)+ ;LOAD OUTBUF WITH DATA TO BE WRITTEN
647 003436 005301 DEC R1 ;FILL OUTBUF
648 003440 001374 BNE XSEABUF ;WITH DATA
649 003442 012737 000061 001172 MOV 061,CMD ;WRITE NO I/E
650 003450 104416 DKCMD ;GO WRITE
651 003452 105777 175354 TSTB 0RSC01 ;CHECK FOR READY
652 003456 100375 BPL -4
653 003460 005777 175346 TST 0RSC01 ;TEST FOR ERROR
654 003464 100010 BPL WRNEXB ;BRANCH IF NO ERROR
655 003466 012737 003420 001010 MOV 0SEABUF,LAD ;SET UP LOOP ADDRESS
656 003474 052737 001000 001126 BIS 0BIT9,FLAG ;SET ERROR BIT IN FLAG
657 003502 104430 LOGW ;LOG WRITE ERROR
658 003504 104034 HLT INCIDA,BA
659 003506 104400 WRNEXB: SCOPE
660 003510 004737 007050 JSR PC,DISBUF ;SET UP NEXT DISK ADDR.
661 003514 007741 BR SEABUF ;WRITE NEXT SECTOR
662 003516 104400 RRDSEC: SCOPE

```

663	003520	104414			RDSECT: ERCLR		;CLEAR ERRORS
664	003522	012737	000071	001172	MOV	071,CMD	;READ NO I/E
665	003530	104416			DKCMD		;DO A READ
666	003532	105777	175274		TSTB	0RSC81	;CHECK FOR READY
667	003536	100375			BPL	.-4	;NOT READY BRANCH BACK
668	003540	005777	175266		TST	0RSC81	;TEST FOR ERROR
669	003544	100006			BPL	ADHGT	;BRANCH IF NO ERROR
670	003546	052737	001000	001126	BIS	0BIT9,FLAG	;SET ERROR FLAG
671	003554	104432			LOGR		;LOG READ ERROR
672	003556	104014			HLT	INCIDA	
673	003560	104400			SCOPE		
674	003562	013702	017416		ADHGT: MOV	OUTBUF,R2	
675	003566	013746	001130		MOV	WRDCT,-(SP)	;SAVE OLD WC
676	003572	004737	006562		JSR	PC,WHTHU	
677	003576	013703	001130		MOV	WRDCT,R3	
678	003602	012637	001130		MOV	(SP)+,WRDCT	;RESTORE OLD WC
679	003606	023712	001134		SANHT: CMP	DMA,(2)	;CMP FOR CORRECT ADDR.
680	003612	001004			BNE	ADERR	;BRANCH IF DATA DID NOT COMPARE
681	003614	005722			TST	(2)+	;GET NEXT ADDRESS OF INBUF
682	003616	005303			DEC	R3	;DEC SECTOR COUNT
683	003620	001372			BNE	SANHT	;TEST NEXT WORD
684	003622	000412			BR	CHKADT	
685	003624	013701	001134		ADERR: MOV	DMA,GOOD	;CORRECT ADDRESS
686	003630	011200			MOV	(2),BAD	;DATA IN ERROR
687	003632	104000			HLT		;DISK ADD DID NOT MATCH WRITTEN ADDRESS
688	003634	104436			LOGC		;LOG COMPARE ERROR
689	003636	004737	014204		JSR	PC,PRNT	;INHIBIT TYPEOUT?
690	003642	001002			BNE	CHKADT	;YES
691	003644	104402	000636		TYPE	,CRLF	
692							
693							
694							
695	003650	104400					
696	003652	004737	007050		CHKADT: SCOPE		
697	003656	000717			JSR	PC,DISBUF	;SET UP NEXT DISK BUFFER
698	003660	013737	001150	001144	BR	RRDSEC	;CHECK NEXT SECTOR
699	003666	042737	000400	001126	MOV	SAVE,SWRDCT	;GET STD WD COUNT
700	003674	032737	100000	001216	BIC	0BIT0,FLAG	;CLEAR TEST FLAG
701	003702	001402			BIT	0BIT15,FLAG3	;DOES OPERATOR WANT TO LOOP ON TEST
702	003704	000137	003326		BEQ	.-6	;NO
					JMP	ADT1A	;YES

*****REPORT ONLY ONE ERROR PER SECTOR*****

703	003710					DATA:		;DATA TEST
704						;		
705						;TEST 2	DATA TEST	
706						;		
707	003710	104400				TST2:	SCOPE	
708	003712	005037	001134			CLR	DMA	;CLEAR DISK ADDRESS
709	003716	104426				CLRDV		;CLEAR RS REGISTERS
710	003720	013737	001144	001130		MOV	SWRDCT,WRDCT	
711	003726	032737	000100	001126		BIT	%BIT6,FLAG	;MULTI PORT?
712	003734	001403				BEG	30	;NO
713	003736	005737	001114			TST	A0B1	;A OR B PORT?
714	003742	001003				BNE	10	;B PORT
715	003744	004737	011350		30:	JSR	PC,APORT	;A PORT
716	003750	000402				BR	20	
717	003752	004737	011416		10:	JSR	PC,BPORT	;B PORT
718	003756	004737	006774		20:	JSR	PC,VECTR	;SETUP INT VECTOR
719	003762	012777	000340	175074		MOV	%340,%STATUS	;SET DISK STATUS REG LOC (206)
720	003770	012737	004022	001152		MOV	%LDAT,HNDER	;SETUP FOR HARD ERROR RETURN
721	003776	013737	017416	001140		MOV	OUTBUF,BUF	;SETUP OUTPUT BUFFER
722	004004	052737	000003	001126		BIS	%3,FLAG	;SET COUNTER TO 3
723	004012	004537	007716			JSR	RS,PASEL	;SET UP DATA BUFFERS
724	004016	005037	001200			LDAT1:	CLR	;CLEAR ERROR FLAG
725	004022	104414			LDAT:	ERCLR		;CLEAR RS REG. IF ERROR
726	004024	004537	006500			JSR	RS,OPDSEL	;SET UP DISK ADDRESS
727	004030	032737	040000	001126		BIT	%BIT14,FLAG	;TEST FOR WRITE
728	004036	001456				BEG	5LH	;TEST FOR READ
729	004040	012737	000161	001172		MOV	%161,CND	;WRITE WITH I/E
730	004046	104416				DKCND		;DO A WRITE
731	004050	004737	011750			JSR	PC,WATT	;WAIT FOR INTERRUPT
732	004054	012737	004022	001010		MOV	%LDAT,LAD	;SETUP SCOPE LOOP
733	004062	032737	001000	001126		BIT	%BIT9,FLAG	;WAS THERE AN ERROR?
734	004070	001423				BEG	WRXBL	;CONT
735	004072	104430				LOGW		;LOG WRITE ERROR
736	004074	005237	001200			INC	LOPCNT	;SET ERROR FLAG
737	004100	004737	014204		20:	JSR	PC,PRNT	;TYPE ?
738	004104	001004				BNE	10	;NO
739	004106	104402	000554			TYPE	,DATA	
740	004112	104402	000564			TYPE	,WRTERR	
741	004116	104044			10:	HLL	%DSIDA	;WRITE ERROR
742	004120	005337	001126			DEC	FLAG	;DEC COUNTER
743	004124	032737	000003	001126		BIT	%3,FLAG	;DONE YET WITH 3RD TRY?
744	004132	001333				BNE	LDAT	;NOT 3 TRIES YET? TRY AGAIN
745	004134	004737	011454			JSR	PC,WTNO	;TYPE CAN NOT WRITE

746	004140	005737	001200		WRXBL:	TST	LOPCNT	;WAS THERE AN ERROR?
747	004144	001402				BEQ	NRX1	;NO
748	004146	004737	011772			JSR	PC,TYPEC	;TYPE RECOVERED
749	004152	005037	001200		NRX1:	CLR	LOPCNT	;CLEAR ERROR FLAG
750	004156	104400				SCOPE		
751	004160	052737	000003	001126		BIS	#3,FLAG	;CLEAR RETRY COUNT
752	004166	004737	007050			JSR	PC,DISBUF	;SET BUFFER FOR WRITE CHECK
753	004172	000711				BR	LDA1	
754	004174	104400			SLH1:	SCOPE		
755	004176	104414			SLH2:	ERCLR		;CLEAR PS REG IF ERRORS
756	004200	004537	006500			JSR	RS,OPDSEL	;IS THE OPERATOR SELECTING THE TRACK
757	004204	032737	020000	001126		BIT	#BIT13,FLAG	;TEST FOR WRITE CHECK
758	004212	001002				BNE	10	;YES
759	004214	000137	004520			JMP	ESH1	;NO
760	004220	013737	017416	001140	10:	MOV	OUTBUF,BUF	;SET UP CURRENT ADDRESS
761	004226	012737	000151	001172		MOV	#151,CMD	;WRITE CHECKWITH I/E
762	004234	104416				DKCMD		;GO WRITE CHECK
763	004236	004737	011750			JSR	PC,WATT	;WAIT FOR INTERRUPT
764	004242	032737	001000	001126	XESH1:	BIT	#BIT9,FLAG	;IS THERE AN ERROR?
765	004250	001505				BEQ	10	;NO ERROR
766	004252	005737	001200			TST	LOPCNT	;1ST ERROR?
767	004256	001001				BNE	20	;NO
768	004260	104434				LOGWC		;YES LOG ERROR
769	004262	032777	000100	174536	20:	BIT	#BIT6,OSNR	;TYPE ALL ERRORS?
770	004270	001007				BNE	30	;YES
771	004272	032777	001000	174526		BIT	#BIT9,OSNR	;LOOP ON ERROR?
772	004300	001003				BNE	30	;YES
773	004302	005737	001200			TST	LOPCNT	;FIRST ERROR?
774	004306	001056				BNE	100	;NO
775	004310	004737	014204		30:	JSR	PC,PRNT	;TYPE OUT?
776	004314	001052				BNE	40	;NO
777	004316	104402	000554			TYPE	,DATA	
778	004322	104402	000574			TYPE	,MCKERR	
779	004326	017702	174506			MOV	ORSDA,R2	;GET CORRECT BA
780	004332	023702	017416			CMP	OUTBUF,R2	;DID A WD GET XFERED?
781	004336	001406				BEQ	90	;NO
782	004340	032777	000400	174460		BIT	#BIT0,OSNR	;XFER MODE?
783	004346	001002				BNE	90	;YES
784	004350	162702	000002			SUB	#2,R2	
785	004354	004737	014204		90:	JSR	PC,PRNT	;TYPEOUT ERRORS?
786	004360	001030				BNE	40	;NO
787	004362	005737	001212			TST	MMAVA	;IS MEN MGMT AVAILABLE?
788	004366	001402				BEQ	70	;NO

789	004370	004737	006626		JSR	PC,PHYCOV	;YES GET VITURAL ADDR
790	004374	010237	001222	78:	MOV	R2,WOPK	;GET BA
791	004400			88:			
792	004400	104402	004404		TYPE	..+2	;ASCIZ <15><12>"(BA)"
793	004414			68:			
794	004414	017746	174602		MOV	0WOPK,-(6)	;PUT 0WOPK ON STACK
795	004420	104404			TYPE0		;TYPE STACK IN OCTAL
796	004422	104402	004426		TYPE	..+2	;ASCIZ "WC"
797	004434	017746	174376		MOV	0RSWC,-(6)	;PUT 0RSWC ON STACK
798	004440	104404			TYPE0		;TYPE STACK IN OCTAL
799	004442	104026		48:	HLT	IDAIDBIBA	;NOTE: BA REG. = +2 OF ACTUAL MEMORY
800							;LOC AFTER WORDS HAVE BEEN XFERED
801	004444	005237	001200	100:	INC	LOPCNT	;INC ERROR COUNT
802	004450	022737	000010	001200	CMP	010,LOPCNT	;10 TRYS YET?
803	004456	001247			BNE	0LH2	;NO
804	004460	004737	006520		JSR	PC,NOREC	;TYPE UNRECOVERABLE
805	004464	005737	001200	18:	TST	LOPCNT	;ANY ERRORS?
806	004470	001402			BEQ	00	;NO
807	004472	004737	011772		JSR	PC,TYPREC	;TYPE RECOVERED
808	004476	005037	001200	58:	CLR	LOPCNT	;CLEAR ERROR COUNTER
809	004502	104400			SCOPE		
810	004504	012737	004176	001010	MOV	0SLH2,LAD	;SETUP LOOP ADDRESS
811	004512	004737	007050		JSR	PC,DISBUF	;SET UP THE DISK BUFFER
812	004516	000422			BR	0LH2A	
813	004520	004537	011302	ESH1:	JSR	R5,CLEAR	;CLEAR BUFFER
814	004524	004537	006500	ESH:	JSR	R5,OPDSEL	;OPERATOR SELECTED DISK ADDRESS?
815	004530	032737	010000	001126	BIT	0BIT12,FLAG	;TEST FOR READ
816	004536	001002			BNE	10	;YES
817	004540	000137	004754		JMP	0STR	;NO READ
818	004544	104400		18:	SCOPE		
819	004546	042737	000003	001126	BIC	03,FLAG	;CLEAR RE-READ COUNT
820	004554	005037	001200		CLR	LOPCNT	;CLEAR FLAG
821	004560	000137	004570		JMP	0SKRD	;CONT
822	004564	000137	004176	0LH2A:	JMP	0LH2	
823	004570	104414		0SKRD:	ERCLR		;CLEAR R5 REG IF ERRORS
824	004572	012737	000171	001172	MOV	0171,CMD	;READ WITH I/E
825	004600	104416			DKCMD		;READ
826	004602	032777	010000	174216	BIT	010000,05NR	;COMPARE?
827	004610	001007			BNE	TAG	;NO
828	004612	032737	000004	001126	BIT	0BIT2,FLAG	;COMPARE?
829	004620	001003			BNE	TAG	;NO
830	004622	004537	010446		JSR	R5,COMPARE	;COMPARE
831	004626	000402			BR	ELH	

832	004630	004737	011750		TAG:	JSR	PC,WATT	;WAIT FOR INTERRUPT
833	004634	032737	001000	001126	ELH:	BIT	0BIT9,FLAG	;IS THERE AN ERROR?
834	004642	001433				BEQ	ADRD	;NO
835	004644	032777	000100	174154		BIT	0BIT6,0SWR	;SOFT ERROR TYPEOUT?
836	004652	001003				BNE	30	;YES
837	004654	005737	001200			TST	LOPCNT	;FIRST ERROR?
838	004660	001011				BNE	20	;NO
839	004662	004737	014204		30:	JSR	PC,PRNT	;TYPEOUT?
840	004666	001004				BNE	10	;NO
841	004670	104402	000554			TYPE	,DATA	
842	004674	104402	000607			TYPE	,RDERR	
843	004700	104432			10:	LOGR		;LOG READ ERROR
844	004702	104044				HLT	1DSIDA	
845	004704	104400			20:	SCOPE		
846	004706	005237	001200			INC	LOPCNT	;COUNT ERRORS
847	004712	022737	000010	001200		CMP	010,LOPCNT	;LAST RETRY?
848	004720	001323				BNE	DSKRD	;NO
849	004722	004737	006520			JSR	PC,NOREC	;TYPE UNRECOVERABLE
850	004726	000137	004746		40:	JMP	ADRI	;CONT
851	004732	104400			ADRI:	SCOPE		
852	004734	005737	001200			TST	LOPCNT	;ANY ERRORS?
853	004740	001402				BEQ	ADRI	;NO
854	004742	004737	011772			JSR	PC,TYPREC	;TYPE RECOVERED
855	004746	004737	007050		ADRI:	JSR	PC,DISBUF	;GO SET UP DISK BUFFER.
856	004752	000434				BR	READCT	;CONT, READING
857	004754	005737	001126		MSRI:	TST	FLAG	
858	004760	100423				BMI	30	;OPERATOR SELECTED PATTERN
859	004762	062737	000002	001136		ADD	02,PATNU	;INC PATTERN INDEX
860	004770	022737	000044	001136		CMP	044,PATNU	
861	004776	001402				BEQ	,+6	
862	005000	000137	003710			JMP	DATAT	;NOT LAST PATTERN EXIT
863	005004	005037	001136			CLR	PATNU	;LAST PATTERN EXIT
864	005010	032777	000002	174010		BIT	0BIT1,0SWR	;DATA TEST ONLY?
865	005016	001006				BNE	20	;YES
866	005020	032737	002000	001126		BIT	0BIT10,FLAG	;DATA TEST ONLY?
867	005026	001404				BEQ	10	;NO
868	005030	005137	001114		30:	COM	ADB1	;ALTERNATE PORTS
869	005034	000137	006066		20:	JMP	EXTPPR	;LOOP
870	005040	000137	005056		10:	JMP	RANEX	;DO NEXT TEST
871								
872	005044	000137	004524		READCT:	JMP	ESH	;CONT, READING
873	005050	052737	100000	001216	RANEL:	BIS	0BIT15,FLAG3	;SET LOOP IN RANDOM TEST GOT
874								;HERE BY STARTING AT LOC 240


```

875 005056
876
877
878
879
880
881
882 005056 104400
883 005060 032737 000004 001126
884 005066 001402
885 005070 000137 006066
886 005074 052737 000400 001126 28:
887 005102 012737 020000 017416
888 005110 013737 017416 001116
889 005116 005737 001212
890 005122 001402
891 005124 005037 177572
892 005130 012737 000042 001136 10:
893 005136 104426
894 005140 012737 176030 001156
895 005146 012737 005706 001152
896 005154 004737 006774
897 005160 012777 000340 173676
898 005166 012737 005270 001010 WRLG1:
899 005174 012737 000001 001222
900 005202 013701 017416
901 005206 004537 010040
902 005212 017737 012200 001134
903 005220 042737 170000 001134
904 005226 052737 000003 001126
905 005234 004737 006562
906 005240 013737 001130 001222 28:
907 005246 013701 017416
908 005252 004537 010040
909 005256 013737 017416 001140
910 005264 005037 001200

RANEX:
;THIS PROGRAM WRITES, WRITECHECKS AND READS 1 SECTOR OF RANDOM DATA FROM RANDOM DISK
;ADDRESSES. THIS TEST WILL MAKE 1000(10) PASSES BEFORE IT IS COMPLETED

;=====
;TEST 3 RANDOM ADDRESS RANDOM DATA TEST
;=====

TST3: SCOPE
BIT 0BIT2,FLAG ;FAST XFER MODE?
BEQ 20 ;NO
JMP EXTPPR ;GET OUT
BIS 0BIT0,FLAG ;SET TEST FLAG
MOV 020000,OUTBUF ;GET STARTING ADDR OF BUF
MOV OUTBUF,VADDR ;SAVE BUFFER ADDR
TST MHAVA ;MEM MGMT AVAILABLE?
BEQ 10 ;NO
CLR 00SR0 ;TURN IT OFF
MOV 042,PATNU ;DO RANDOM COMPARE
CLR DV ;INIT DRIVE
MOV 0-1000,,PASSC ;SET UP PASS COUNT
MOV 0WRRED,HRDER ;SET UP FOR HARD ERROR
JSR PC,VECTRR ;SETUP INTERRUPT VECTOR
MOV 0340,0STATUS
WRLG1: MOV 0WRERR,LAD ;SETUP LOOP ADDRESS
MOV 01,WORK ;SET UP RANDOM GENERATOR WORD
MOV OUTBUF,R1
JSR RS,RANDOM ;GENERATE RANDOM DATA
MOV 0OUTBUF,DMA ;SET UP DISK ADDRESS
BIC 0170000,DMA
BIS 03,FLAG ;SET COUNTER
JSR PC,WHTHU ;GET WORD COUNT
MOV WRDCT,WORK ;GENERATE RANDOM BUFFER
MOV OUTBUF,R1
JSR RS,RANDOM
MOV OUTBUF,BUF ;SET UP OUTPUT BUFFER
CLR LOPCNT ;CLR ERROR FLAG

```

911	005270	104414			WRERR:	ERCLR		
912	005272	012737	000161	001172		MOV	0161,CMD	;WRITE WITH I/E
913	005300	104416				DKCMD		;WRITE
914	005302	004737	011750			JSR	PC,WATT	;WAIT FOR INTERRUPT
915	005306	032737	001000	001126	20:	BIT	0BIT9,FLAG	;WAS THERE AN ERROR?
916	005314	001435				BEO	WRRCK1	;NO
917	005316	032777	000100	173502		BIT	0BIT6,0SWR	;TYPE RETRY ?
918	005324	001003				BNE	50	;YES
919	005326	005737	001200			TST	LOPCNT	;FIRST TIME?
920	005332	001013				BNE	60	;NO
921	005334	104430			50:	LOGW		;LOG WRITE ERROR
922	005336	005237	001200			INC	LOPCNT	;SET ERROR FLAG
923	005342	004737	014204			JSR	PC,PRNT	;TYPEOUT?
924	005346	001004				BNE	30	;YES
925	005350	104402	000716			TYPE	,RANDM	
926	005354	104402	000564			TYPE	,WRTERR	
927	005360	104044			30:	HLT	1DSIDA	
928	005362	104400			60:	SCOPE		
929	005364	005337	001126			DEC	FLAG	
930	005370	032737	000003	001126		BIT	03,FLAG	
931	005376	001334				BNE	WRERR	;RETRY
932	005400	004737	011434			JSR	PC,WTHO	;TYPE CAN NOT WRITE
933	005404	000137	006010			JMP	EXRAX	;GET NEW NUMBER

934	005410	005737	001200			WRRCK1: TST	LOPCNT	;ANY ERRORS?
935	005414	001402				BEQ	18	;NO
936	005416	004737	011772			JSR	PC,TYPEC	;TYPE RECOVERED
937	005422	104400			18:	SCOPE		
938	005424	005037	001200			CLR	LOPCNT	;CLEAR LOOP COUNT
939	005430	104414				WRRCK1: ERCLR		;CLEAR RS REG IF ERRORS
940	005432	012737	000151	001172		MOV	0151,CMD	;WRITE CHECK WITH I/E
941	005440	104416				DKCMD		;WRITE CHECK
942	005442	004737	011750			JSR	PC,WATT	;WAIT FOR INTERRUPT
943	005446	032737	001000	001126	48:	BIT	0BIT0,FLAG	;ERROR?
944	005454	001453				BEQ	18	;NO
945	005456	032777	000100	173342		BIT	0BIT6,08MR	;TYPE ALL RETRY?
946	005464	001003				BNE	20	;YES
947	005466	005737	001200			TST	LOPCNT	;FIRST ERROR?
948	005472	001030				BNE	50	;NO
949	005474	104434			28:	LOGWC		;LOG WRITE CK
950	005476	004737	014204			JSR	PC,PRNT	;TYPEOUT?
951	005502	001052				BNE	60	;NO
952	005504	104402	000716			TYPE	,RANDM	
953	005510	104402	000574			TYPE	,WCKERR	
954	005514	017737	173320	001222		MOV	0RSBA,WORK	;GET CORRECT BA
955	005522	162737	000002	001222		SUB	02,WORK	
956	005530	104402	005534			TYPE	,+2	;ASCIZ <15><12>*(BA)=
957	005544	017746	173452			MOV	0WORK,-(6)	;PUT 0WORK ON STACK
958	005550	104404				TYPE0		;TYPE STACK IN OCTAL
959	005552	104026				HLT	1DBIDA,BA	;BA=MEMORY LOC +2 OF ACTUAL WORD
960	005554	005237	001200		58:	INC	LOPCNT	;INC RETRY COUNT
961	005560	022737	000010	001200		CMP	010,LOPCNT	;LAST ONE YET?
962	005566	001320				BNE	WRRCK	;NO
963	005570	104402	000522			TYPE	,UNRECO	
964	005574	005037	001200			CLR	LOPCNT	;CLEAR LOPCNT
965	005600	000137	006010			JMP	EXRAX	;GET NEW NUMBER
966	005604	005737	001200		18:	TST	LOPCNT	;ANY ERRORS?
967	005610	001407				BEQ	60	;NO
968	005612	104402	000616			TYPE	,RECOV	
969	005616	013746	001200			MOV	LOPCNT,-(6)	;GET NUMBER
970	005622	104406				TYPE8		;TYPE IT
971	005624	104402	000636			TYPE	,CRLF	
972	005630	104400			68:	SCOPE		
973	005632	052737	000003	001126		BIS	03,FLAG	;SET COUNTER

974	005640	104400				SCOPE		
975	005642	005037	001200			CLR	LOPCNT	;CLEAR COUNTER
976	005646	004537	011302			JSR	RS,CLEAR	;CLEAR BUFFER
977	005652	104414			RREAD:	ERCLR		;CLEAR RS REG IF ERRORS
978	005654	012737	000171	001172		MOV	0171,CMD	;READ WITH I/E
979	005662	104416				DKCMD		;READ
980	005664	032777	010000	173134		BIT	0BIT12,08WR	;COMPARE ?
981	005672	001003				BNE	TAG1	;NO
982	005674	004537	010446			JSR	RS,COMPARE	;YES
983	005700	000402				BR	RWRED	;CONT
984	005702	004737	011750		TAG1:	JSR	PC,WATT	;WAIT FOR INTERRUPT
985	005706	032737	001000	001126	RWRED:	BIT	0BIT0,FLAG	;IS THERE AN ERROR?
986	005714	001438				BEQ	EXRAX	;NO
987	005716	104432				LOGR		;LOG READ ERR
988	005720	032777	000100	173100	10:	BIT	0BIT6,08WR	;TYPE ALL ERRORS?
989	005726	001016				BNE	20	;YES
990	005730	032777	001000	173070		BIT	0BIT0,08WR	;LOOP ON ERROR?
991	005736	001012				BNE	20	;YES
992	005740	005737	001200			TST	LOPCNT	;FIRST ERROR?
993	005744	001010				BNE	30	;NO
994	005746	004737	014204			JSR	PC,PRNT	;TYPEOUT?
995	005752	001004				BNE	20	;NO
996	005754	104402	000716			TYPE	,RANDM	
997	005760	104402	000607			TYPE	,RDERR	
998	005764	104006			20:	HLT	IDBIDA	
999	005766	104400			30:	SCOPE		
1000	005770	005237	001200			INC	LOPCNT	;UPDATE COUNTER
1001	005774	022737	000010	001200		CMP	010,LOPCNT	;LAST TRY YET?
1002	006002	001323				BNE	RREAD	;RETRY
1003	006004	004737	006520			JSR	PC,NOREC	;TYPE UNRECOVERABLE
1004	006010	005737	001200		EXRAX:	TST	LOPCNT	;ANY ERRORS?
1005	006014	001402				BEQ	EXRXX	;NO
1006	006016	004737	011772			JSR	PC,TYPEC	;TYPE RECOVERED
1007	006022	104400			EXRXX:	SCOPE		
1008	006024	005237	001156			INC	PASSC	;+1 PASS COUNT
1009	006030	001402				BEQ	10	;IS TEST DONE?
1010	006032	000137	005166			JMP	WRLG1	;NO
1011	006036	005037	001136		10:	CLR	PATNU	;END OF TEST
1012	006042	042737	000400	001126		BIC	0BIT0,FLAG	;CLEAR TEST FLAG
1013	006050	032737	100000	001216		BIT	0BIT15,FLAG3	;LOOP ON THIS TEST?
1014	006056	001402				BEQ	,+6	;NO
1015	006060	000137	005056			JMP	RANEX	;YES


```

1016 ;CHECK FOR MULTI DISK MODE
1017 ;IF IN MULTI DISK MODE REPORT "END"
1018 ;IF LAST DISK ON SYSTEM HAS BEEN EXERCISED.
1019
1020 ;*****
1021 ;TEST 4 TEST FOR MULTI DISK MODE
1022 ;*****
1023 TST4: SCOPE
1024 EXTPPR: CLR DMA
1025 CLRDV ;INIT DRIVE
1026 BIT #BIT11,FLAG ;ARE WE IN MULTI DISK MODE
1027 BEQ EXTPP ;NO REPORT "END"
1028 JSR PC,TRYNX ;YES TEST FOR ALL DRIVES
1029 JMP EXNPLG ;RESTART TESTING OF DRIVES
1030 EXTPP: JSR PC,DONEE ;GET PASS COUNT
1031
1032 ;THIS ROUTINE CLEARS THE DRIVE
1033 ;REGISTERS IF THERE WAS AN ERROR
1034
1035 .ERCLR: BIT #BIT9,FLAG ;ANY ERRORS?
1036 BEQ 10 ;NO
1037 CLRDV ;CLEAR ALL ERRORS
1038 BIC #BIT9,FLAG ;CLEAR ERROR FLAG
1039 10: RTI ;EXIT
1040
1041 ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
1042 ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
1043 ;BYTE OF THE CONTROL REGISTER.
1044
1045 .DKCMD: MOV DMA,ORSDA ;LOAD DISK ADD
1046 CLR INTPLG ;CLEAR INTERRUPT FLAG
1047 BIT #BIT13,FLAG2 ;IN MAPPING ROUTINE?
1048 BNE 40 ;YES
1049 BIT #BIT2,FLAG ;MAX DATA TEST?
1050 BEQ 40 ;NO
1051 CMP #42,PATNU ;RANDOM DATA?
1052 BEQ 40 ;YES
1053 BIS #10,ORSC62 ;SET BAI
1054 CLR ORSWC ;64K XFER
1055 MOV #OUTBUF,OR8BA ;SETUP BA
1056 BR 20 ;CONT
1057 MOV BUF,OR8BA ;LOAD (CMA) BUSS ADDRESS
1058 MOV WRDCT,R2 ;SET NEGATIVE
1059 NEG R2 ;WORD COUNT
1060 MOV R2,ORSWC ;LOAD WC
1061 BIT #BIT0,FLAG ;RANDOM TEST?
1062 BNE 10 ;YES A PORT ONLY WITH NO MEM NGMT
1063 TST MNAVA ;MEM NGMT AVAILIABLE?
1064 BEQ 20 ;NO
1065 BIT #BIT5,FLAG2 ;SET A17 IN RSC61
1066 BEQ 30 ;NO
1067 BIS #BIT9,CMD ;YES

```


1068	006302	032737	000020	001122	30:	BIT	0BIT4,FLAG2	;SET A16?
1069	006310	001403				BEQ	20	;NO
1070	006312	052737	000400	001172		BIS	0BIT0,CMD	;YES
1071	006320	032737	000100	001126	20:	BIT	0BIT6,FLAG	;MULTI PORT?
1072	006326	001406				BEQ	10	;NO
1073	006330	005737	001114			TST	A0B1	;TEST A OR B PORT?
1074	006334	001403				BEQ	10	;A PORT
1075	006336	052737	002000	001172		BIS	0BIT10,CMD	;B PORT
1076	006344	013777	001172	172460	10:	MOV	CMD,0RSC01	;LOAD FUNCTION REG.
1077	006352	000002				RTI		;RETURN FROM TRAP
1078								
1079								
1080								
1081								
1082	006354	042737	001000	001126	DKINT:	BIC	0BIT9,FLAG	;CLEAR ERROR BIT
1083	006362	005777	172444			TST	0RSC01	;TEST FOR ERROR
1084	006366	100401				BMI	20	
1085	006370	000425				BR	INTEXT	;JUMP IF NO ERRORS
1086	006372	017702	172434		20:	MOV	0RSC01,R2	;GET CONTENTS OF C01
1087	006376	042702	037777			BIC	037777,R2	;CLEAR ALL BUT SC AND TRE
1088	006402	022702	140000			CMP	0140000,R2	;IS SC AND TRE BOTH SET?
1089	006406	001413				BEQ	TRUERR	;YES THERE IS SOME KIND OF XFER ERROR
1090	006410	032777	100000	172426		BIT	0100000,0RSC01	;IS THE ATA BIT SET?
1091	006416	001007				BNE	TRUERR	;YES
1092	006420	104140				HLT	1AS100	;WRONG UNIT INTERRUPTED
1093								;IF YOU HAVE JUST POWERED UP A DRIVE OR
1094								;ARE RUNNING THE POWER FAIL TEST,
1095								;INTERRUPTS WILL OCCUR FROM DRIVES OTHER
1096								;THAN THE UNIT UNDER TEST. IF THIS TIMEOUT
1097								;SHOWS NO ERRORS IN THE REGISTERS OF THE DRIVE
1098								;UNDER TEST, THAT DRIVE IS OK
1099	006422	012777	177777	172420	10:	MOV	0-1,0RSC01	;CLEAR ALL ATA BITS
1100	006430	013716	001152			MOV	NRDR,(SP)	;GET RETURN ADD.
1101	006434	000002				RTI		;RETRY
1102	006436	052737	001000	001126	TRUERR:	BIS	0BIT9,FLAG	;SET ERROR BIT
1103	006444	032777	004000	172354	INTEXT:	BIT	0BIT11,0RSC01	;HALT ON COMPLETION FLAG
1104	006452	001401				BEQ	.+4	
1105	006454	000000				HALT		;YES BIT 11 SET IN SWR HALT
1106	006456	032737	002000	001122		BIT	0BIT10,FLAG2	;WAIT IN BACKGROUND TEST?
1107	006464	001402				BEQ	10	;NO
1108	006466	012716	012416		20:	MOV	0NRPRRT,(SP)	;MODIFY RETURN ADD.
1109	006472	010637	001176		10:	MOV	SP,INTFLG	;SET INT FLG
1110	006476	000002				RTI		;EXIT
1111								
1112								
1113								
1114								
1115	006500	032737	000040	001126	OPDSEL:	BIT	0BITS,FLAG	;OPTIONAL DMA?
1116	006506	001403				BEQ	10	;NO
1117	006510	013737	001142	001134		MOV	TDNA,DNA	;GET OPT, DMA
1118	006516	000205			10:	RTS	R5	;EXIT

;ROUTINE TO SET UP TRACK 0 FROM OPTION
;ENTER FROM JSR R5, OPDSEL


```

1119
1120
1121          .EVEN
1122
1123 006520 004737 014204      NOREC: JSR      PC,PRNT      ;TYPEOUT?
1124 006524 001002              BNE      10              ;NO

1125 006526 104402 000522      10:      TYPE      ,UNRECO
1126 006532 005037 001200      CLR      LOPCNT      ;CLEAR LOOP COUNTER
1127 006536 000207      RTS      PC
1128
1129 006540 052737 040000 001122 FNDTYP: BIS      @BIT14,FLAG2 ;SET CHECK DRIVE TYPE FLAG
1130 006546 004737 001746      JSR      PC,UNTP      ;CHECK DRIVE TYPE FLAG
1131 006552 042737 040000 001122      BIC      @BIT14,FLAG2 ;CLEAR DRIVE TYPE FLAG
1132 006560 000207      RTS      PC
1133 006562 022737 000004 001166 WHTNU: CMP      @4,RS04DT ;RS03LA?
1134 006570 001004      BNE      10              ;NO
1135 006572 012737 000040 001130      MOV      @40,WRDCT ;GET WORD COUNT
1136 006600 000411      BR      20
1137 006602 012737 000200 001130 10:      MOV      @200,WRDCT ;RS04
1138 006610 005737 001166      TST      RS04DT ;RS04?
1139 006614 001003      BNE      20              ;YES
1140 006616 012737 000100 001130      MOV      @100,WRDCT ;NO
1141 006624 000207      RTS      PC
1142
1143

```

```

1144          ;ROUTINE TO CALCULATE VITURAL ADDR
1145
1146          PHYCOV: SWAB      R2          ;CALCULATE FROM PHYSICAL ADDR
1147          JSR      PC,PRR2
1148          ROR      R2
1149          BIC      @177770,R2          ;GET REG 0
1150          BIT      @BIT0,ORSCB1        ;IS A16 SET?
1151          BEQ      10              ;NO
1152          BIS      @BIT3,R2            ;YES
1153          BIT      @BIT9,ORSCB1        ;IS A17 SET?
1154          BEQ      20              ;NO
1155          BIS      @BIT4,R2            ;YES
1156          MOV      STANEM,WORK1        ;GET BANK 0 FOR -A- PORT
1157          TST      AOB1              ;ARE WE ON -A- PORT?
1158          BEQ      30              ;YES
1159          MOV      STBCOM,WORK1        ;NO -B- PORT
1160          SUB      WORK1,R2            ;GET STARTING BANK 0
1161          ADD      @1,R2              ;GET OFFSET FOR REG 0
1162          SWAB      R2                ;GET BANK 0 INTO
1163          ROL      R2                ;UPPER BITS
1164          ROL      R2
1165          ROL      R2
1166          ROL      R2
1167          ROL      R2
1168          MOV      @R2BA,WORK1        ;GET OFFSET FOR ADDR IF ANY
1169          SUB      @2,WORK1            ;CORRECT IT
1170          BIC      @160000,WORK1       ;CLEAR JUNK
1171          BIS      R2,WORK1            ;GET REG NO
1172          MOV      WORK1,R2
1173          RTS      PC
1174
1175          VECTRR: MOV      @DKINT,ORSVEC ;SETUP INTERRUPT VECTORS
1176          MOV      PRIORITY,PS        ;PRZORITY 4
1177          RTS      PC
1178
1179          ;THIS ROUTINE IS USED FOR DELAYING THE START OF THIS PROGRAM
1180          ;IF POWER FAILED DURING TESTING. THIS WILL GIVE THE DRIVES TIME TO GET UP
1181          ;TO SPEED. THE DELAY WILL BE ABOUT 3-5 MINUTES DEPENDING UPON THE PROCESSOR
1182
1183          TIMUP: MOV      @677,WORK    ;677,WORK
1184          MOV      @177777,WORK1      ;177777,WORK1
1185          NOP
1186          DEC      WORK1
1187          BNE      20
1188          DEC      WORK
1189          BNE      10
1190          JMP      ADTST

```



```

1191 ;ROUTINE TO SETUP DISK BUFFERS
1192 ;ADD WORD COUNT TO STARTING DISK ADDRESSES
1193 ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
1194
1195 007050 032737 000040 001126 DISBUF: BIT 0BITS,FLAG ;DID OPERATOR SELECT PATTERNS
1196 007056 001402 BEQ 28 ;NO
1197 007060 000137 007342 JMP BUFEXIT ;YES
1198 007064 022737 000042 001136 28: CMP 042,PATNU ;RANDOM PATTERN?
1199 007072 001443 BEQ 10 ;YES
1200 007074 032737 000004 001126 BIT 0BIT2,FLAG ;MAX TST?
1201 007102 001437 BEQ 10 ;NO
1202 007104 022737 000004 001166 CMP 04,R804DT ;R803LA?
1203 007112 001010 BNE 40 ;NO
1204 007114 022737 004000 001134 CMP 04000,DNA ;DONE YET?
1205 007122 001507 BEQ BUFEXIT ;YES
1206 007124 062737 004000 001134 ADD 04000,DNA ;UPDATE DNA
1207 007132 000207 RTS PC
1208 007134 005737 001166 40: TST R804DT ;R804?
1209 007140 001010 BNE 30 ;YES
1210 007142 022737 006000 001134 CMP 06000,DNA ;R803
1211 007150 001474 BEQ BUFEXIT ;DONE GET OUT
1212 007152 062737 002000 001134 ADD 02000,DNA ;UPDATE DNA
1213 007160 000207 RTS PC ;EXIT
1214 007162 022737 007000 001134 30: CMP 07000,DNA ;DONE YET?
1215 007170 001464 BEQ BUFEXIT ;YES
1216 007172 062737 001000 001134 ADD 01000,DNA ;UPDATE ADDR
1217 007200 000207 RTS PC
1218 007202 004737 007556 10: JSR PC,BLSE ;DEFINE BLOCK SIZE
1219 007206 013737 001154 001224 MOV BLOCK,WORK1
1220 007214 005237 001134 INCSEC: INC DNA ;+1 SECTOR COUNT
1221 007220 022737 010000 001134 CMP 010000,DNA ;DONE YET?
1222 007226 001445 BEQ BUFEXIT ;YES
1223 007230 005337 001154 DEC BLOCK ;-1 FROM BLOCK COUNT
1224 007234 001401 BEQ CONDAE ;CMP DNA TO RSDA
1225 007236 000766 BR INCSEC ;RECTCLE
1226 007240 032737 001000 001126 CONDAE: BIT 0BIT9,FLAG ;ANY ERRORS?
1227 007246 001401 BEQ 10 ;NO ERRORS DO COMPARE ON RSDA
1228 007250 000207 RTS PC ;ERRORS DO NOT COMPARE RSDA
1229 007252 023777 001134 171562 10: CMP DNA,0RSDA ;COMPARE RSDA WITH DNA
1230 007260 001425 BEQ CMDAE ;SHOULD BE EQUAL
1231 007262 104432 LOGR ;AFTER TRANSFER RSDA AND DNA SHOULD BE =
1232 ;IF NOT, RSDA IS NOT CORRECT, DNA CONTAINS
1233 ;WHAT RSDA SHOULD =
1234 007264 013701 001134 MOV DNA,GOOD ;GET DNA FOR CORRECT AND IN GOOD
1235 007270 017700 171546 MOV 0RSDA,BAD ;GET RSDA INTO BAD
1236 007274 104000 HLT ;RSDA=BAD DNA=GOOD SEE COMMENTS 7 LINES ABOVE
1237 007276 004737 014204 JSR PC,PRNT ;TYPEOUT?
1238 007302 001014 BNE CMDAE ;NO
1239 007304 011637 001222 MOV (SP),WORK ;GET TEST PC FROM WHERE IT CAME
1240 007310 104402 007314 TYPE ,,+2 ;ASCIZ "TST PC="
1241 007326 013746 001222 MOV WORK,-(6) ;PUT WORK ON STACK
1242 007332 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
1243 007334 105737 001126 CMDAE: TSTB FLAG ;LAST DISK BUFFER?
1244 007340 100032 BPL BUFINX ;NO

```


1245	007342	005037	001134		BUFEXIT:CLR	DNA	;CLEAR ADDRESS BITS LAST DISK BUFFER
1246	007346	062716	000002		ADD	02,(6)	;INC STOCK POINTER
1247	007352	042737	000200	001126	AKH: BIC	0200,FLAG	;CLEAR LAST DISK BUFFER FLAG
1248	007360	032737	000400	001126	BIT	0BIT0,FLAG	;RANDOM TEST OR ADDR TEST?
1249	007366	001404			BEO	10	;NO
1250	007370	013737	001144	001130	28: MOV	SWRDCI,WRDCI	
1251	007376	000466			BR	EXTDR	;EXIT
1252	007400	032737	000100	001126	18: BIT	0BIT6,FLAG	;MULTI PORT?
1253	007406	001770			BEO	20	;NO
1254	007410	005737	001114		TST	A0B1	;A OR B PORT?
1255	007414	001765			BEO	20	;A PORT
1256	007416	013737	001112	001130	MOV	WDCT0,WRDCI	;B PORT
1257	007424	000453			BR	EXTDR	;GET OUT
1258	007426	005037	001226		BUFINX: CLR	WORK2	;CLEAR WORK2 FOR BLOCK COUNTER
1259	007432	013702	001134		MOV	DNA,R2	;PUT WORKING DISK ADD INTO WORK
1260	007436	005237	001226		XINCSIC: INC	WORK2	;INCREMENT BLOCK COUNT
1261	007442	022702	007777		CMP	07777,R2	;CMP FOR LAST SECTOR
1262	007446	001405			BEO	XINCSUR	;+1 SURFACE LAST SECTOR BRANCH
1263	007450	005202			INC	R2	;INC DNA
1264	007452	005337	001224		DEC	WORK1	;DEC BLOCK COUNT
1265	007456	001367			BNE	XINCSIC	;FILLED STANDARD BUFFER YET?
1266	007460	000734			BR	AKH	;WILL TAKE STANDARD SIZE WORD COUNT
1267	007462	013737	001226	001130	XINCSUR: MOV	WORK2,WRDCI	;SETTING UP BLOCK COUNT
1268	007470	000241			CLC		;FOR NON STANDARD BUFFER SIZE
1269	007472	006137	001130		ROL	WRDCI	
1270	007476	006137	001130		ROL	WRDCI	
1271	007502	006137	001130		ROL	WRDCI	
1272	007506	006137	001130		ROL	WRDCI	
1273	007512	006137	001130		ROL	WRDCI	
1274	007516	022737	000004	001166	CMP	04,R804DT	;R803LA?
1275	007524	001410			BEO	10	;YES
1276	007526	000241			CLC		
1277	007530	006137	001130		ROL	WRDCI	
1278	007534	005737	001166		TST	R804DT	;R804?
1279	007540	001402			BEO	10	;NO
1280	007542	006137	001130		ROL	WRDCI	;YES
1281	007546	052737	000200	001126	18: BIS	0200,FLAG	;SET LAST DISK BUFFER FLAG
1282	007554	000207			EXTDR: RTS	PC	;EXIT


```

1203 ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
1204 007556 022737 000004 001166 BLSZ: CMP 04,R804DT ;R803LA
1205 007564 001004 BNE 38 ;NO
1206 007566 012737 000037 001154 MOV 037,BLOCK ;YES
1207 007574 000411 BR 28 ;CONTINUE
1208 007576 012737 000177 001154 38: MOV 0177,BLOCK ;SETUP FOR R804
1209 007604 005737 001166 TST R804DT ;R804?
1210 007610 001003 BNE 28 ;YES
1211 007612 012737 000077 001154 18: MOV 077,BLOCK ;PUT SECTOR SIZE INTO BLOCK
1212 007620 013702 001130 28: MOV WRDCT,R2 ;FETCH WORD COUNT
1213 007624 033702 001154 BIT BLOCK,R2 ;ARE THEY EQUAL?
1214 007630 001406 BEQ RORBLK ;YES
1215 007632 043702 001154 BIC BLOCK,R2 ;SET UP BLOCK OVERFLOW
1216 007636 005237 001154 INC BLOCK
1217 007642 063702 001154 ADD BLOCK,R2
1218 007646 000241 RORBLK: CLC
1219 007650 006002 ROR R2
1220 007652 022737 000004 001166 CMP 04,R804DT ;R803LA
1221 007660 001003 BNE 28 ;NO
1222 007662 004737 011736 JSR PC,RRR2 ;YES
1223 007666 000410 BR 18 ;CONTINUE
1224 007670 000241 28: CLC
1225 007672 006002 ROR R2
1226 007674 004737 011736 JSR PC,RRR2
1227 007700 005737 001166 TST R804DT ;R804?
1228 007704 001401 BEQ 18 ;NO
1229 007706 006002 ROR R2 ;YES
1230 007710 010237 001154 18: MOV R2,BLOCK ;BLOCK COUNT
1231 007714 000207 RTS PC ;EXIT
1232
1233 ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
1234 ;ENTER FROM JSR RS,PASEL
1235 007716 012737 010352 000004 PASEL: MOV 00EN,004 ;SETUP TRAP
1236 007724 012737 000340 000006 NOV 0340,006 ;VECTOR
1237 007732 013700 001136 NOV PATNU,R0 ;SET UP PATTERN NUMBER
1238 007736 010003 NOV R0,R3 ;GET PATTERN 0
1239 007740 000241 CLC ;MAKE IT 0
1240 007742 006003 ROR R3 ;TO PATTERN 0 IN LISTING
1241 007744 010377 171060 NOV R3,0DISPLAY ;DISPLAY PATTERN 0
1242 007750 013737 001130 001222 NOV WRDCT,WORK ;SET UP WORK
1243 007756 013701 001116 NOV VADDR,R1 ;LOC. OF OUTBUFFER
1244 007762 022700 000042 18: CMP 042,R0 ;TEST FOR RANDOM DATA NUMBER
1245 007766 001424 BEQ RANDOM ;GO GENERATE RANDOM DATA
1246 007770 032737 000004 001126 BIT 0BIT2,FLAG ;MAX TEST?
1247 007776 001404 BEQ 28 ;NO
1248 010000 016037 000254 017416 NOV PAT0(0),OUTBUF ;GET PATTERN
1249 010006 000205 RTS R5
1250 010010 016000 000254 28: NOV PAT0(0),R0
1251 010014 010021 FILDAT: NOV R0,(1)+ ;FILL BUFFER
1252 010016 005337 001222 DEC WORK ;DEC. WORK COUNT
1253 010022 001374 BNE FILDAT ;LOAD NEXT WORD
1254 010024 012737 000006 000004 PASEX: NOV 06,004 ;RESTORE
1255 010032 005037 000006 CLR 006 ;TRAP
1256 010036 000205 RTS R5 ;BUFFER FULL

```



```

1337                                ;RANDOM DATA GENERATOR SUBROUTINE
1338
1339 010040 013737 010204 010210 RANDOM: MOV LONUM,LOSAY
1340 010046 013737 010206 010212 MOV HINUM,HISAY
1341 010054 013700 010204 RAND1: MOV LONUM,R0 ;SET UP R0 WITH 5 DIGITS LOW
1342 010060 013704 010206 MOV HINUM,R4 ;SET UP R1 WITH 5 DIGITS HIGH
1343 010064 012703 000007 MOV #7,R3 ;SET UP SHIFT COUNT
1344 010070 005002 CLR R2 ;CLEAR R2
1345 010072 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
1346 010074 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO
1347 010076 006102 ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2
1348 010100 005303 DEC R3 ;DECREMENT R3
1349 010102 001373 BNE SHIFT ;CONTINUE SHIFT LOOP
1350 010104 063700 010204 ADD LONUM,R0 ;ADDN IN NUMBER TO MAKE X 129
1351 010110 005504 ADC R4 ;PROPOGATE CARRY
1352 010112 063704 010206 ADD HINUM,R4 ;ADDN IN NUMBER TO MAKE X 129
1353 010116 005502 ADC R2 ;PROPOGATE CARRY
1354 010120 062700 001057 ADD #1057,R0 ;ADDN LOW CONSTANT
1355 010124 005504 ADC R4 ;PROPOGATE CARRIES
1356 010126 005502 ADC R2 ;PROPOGATE AGAIN
1357 010130 062704 047401 ADD #47401,R4 ;ADDN HIGH CONSTANT
1358 010134 005502 ADC R2 ;PROPOGATE CARRY
1359 010136 062702 000006 ADD #6,R2 ;ADDN HIGHEST CONSTANT
1360 010142 062700 000002 ADD #2,R0 ;REPRIME R0 WITH HIGH DIGIT
1361 010146 005504 ADC R4 ;PROPOGATE CARRY
1362 010150 010037 010204 MOV R0,LONUM ;PUT R0 BACK IN LONUM
1363 010154 010021 001222 MOV R0,(1)+ ;HOLD LONUM FOR PROGRAM
1364 010156 005337 DEC WORK
1365 010162 001406 BEQ EXGEN
1366 010164 010437 010206 MOV R4,HINUM ;PUT R1 BACK IN HINUM
1367 010170 010421 001222 MOV R4,(1)+ ;HOLD HINUM FOR PROGRAM
1368 010172 005337 DEC WORK
1369 010176 001326 BNE RAND1
1370 010200 000137 010024 EXGEN: JMP PAGEX ;RETURN TO PROGRAM
1371 010204 000000 LONUM: 0
1372 010206 000000 HINUM: 0
1373 010210 000000 LOSAY: 0
1374 010212 000000 HISAY: 0
1375
1376 010214 013737 001214 001144 RESTOR: MOV SAVWC,SWRDCT ;RESTORE ORIGINAL
1377 010222 013737 001144 001130 MOV SWRDCT,WRDCT ;WORD COUNT
1378 010230 013737 001220 001112 MOV SAVWCB,WDCTB
1379 010236 000205 RTS R5

```



```

1300 ;RANDOM DATA GENERATOR SUBROUTINE
1301 ;WHEN SWITCH = 0 WE COME HERE
1302
1303 RAND:  MOV     LONUM1,R0      ;SET UP R0 WITH 5 DIGITS LOW
1304         MOV     HINUM1,R4      ;SET UP R1 WITH 5 DIGITS HIGH
1305         MOV     R7,R3          ;SET UP SHIFT COUNT
1306
1306         CLR     R2              ;CLEAR R2
1307 SHIFT1: ASL     R0              ;SHIFT R0 LEFT AND
1308         ROL     R4              ;ROTATE CARRY INTO LSB OF R1 INTO
1309         ROL     R2              ;ROTATE CARRY OUT OF R1 INTO R2
1310         DEC     R3              ;DECREMENT R3
1311         BNE     SHIFT1          ;CONTINUE SHIFT LOOP
1312         ADD     LONUM1,R0      ;ADDN IN NUMBER TO MAKE X 129
1313         ADC     R4              ;PROPAGATE CARRY
1314         ADD     HINUM1,R4      ;ADDN IN NUMBER TO MAKE X 129
1315         ADC     R2              ;PROPAGATE CARRY
1316         ADD     R1057,R0       ;ADDN LOW CONSTANT
1317         ADC     R4              ;PROPAGATE CARRIES
1318         ADC     R2              ;PROPAGATE AGAIN
1319         ADD     R47401,R4      ;ADDN HIGH CONSTANT
1320         ADC     R2              ;PROPAGATE CARRY
1321         ADD     R6,R2           ;ADDN HIGHEST CONSTANT
1322         ADD     R2,R0          ;REPRIME R0 WITH HIGH DIGIT
1323         ADC     R4              ;PROPAGATE CARRY
1324         MOV     R0,LONUM1      ;PUT R0 BACK IN LONUM
1325         MOV     R4,HINUM1      ;PUT R1 BACK IN HINUM
1326 EXGEN1: RTS     R5              ;RETURN TO PROGRAM
1327 LONUM1: 0
1328 HINUM1: 0
1329
1330 ;TRAP OUT ROUTINE WHEN CREATING DATA BUFFER
1331
1332 NEW:
1333         TYPE    ..+2           ;.ASCIZ <15><12>"NIN PORT "
1334         TST     A001           ;FIND WHAT DATA BUFFER
1335         BNE     30             ;BRANCH IF 0
1336         TYPE    ..+2           ;.ASCIZ "A"
1337         BR      40
1338
1339 30:      TYPE    ..+2           ;.ASCIZ "B"
1340         MOV     R6,R04         ;RESTORE
1341         CLR     R06            ;TRAP
1342         BIT     RBIT15,08WR    ;HALT?
1343         BEQ     20             ;NO
1344         HALT
1345 20:      JMP     00BEGIN
1346
1347 1400 010254 005002
1348 1401 010256 006300
1349 1402 010260 006104
1350 1403 010262 006102
1351 1404 010264 005303
1352 1405 010266 001373
1353 1406 010270 063700 010346
1354 1407 010274 005504
1355 1408 010276 063704 010350
1356 1409 010302 005502
1357 1410 010304 062700 001057
1358 1411 010310 005504
1359 1412 010312 005502
1360 1413 010314 062704 047401
1361 1414 010320 005502
1362 1415 010322 062702 000006
1363 1416 010326 062700 000002
1364 1417 010332 005504
1365 1418 010334 010037 010346
1366 1419 010340 010437 010350
1367 1420 010344 000205
1368 1421 010346 000000
1369 1422 010350 000000
1370 1423
1371 1424
1372 1425 010442 000137 001234

```



```

1426                                     ;THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
1427                                     ;ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 4 OF THE SWITCH
1428                                     ;REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN ERROR HAS BEEN
1429                                     ;FOUND AND WILL REPORT UP TO 8 VERIFY ERRORS WITHIN THE SAME INPUT OPERATION.
1430
1431 010446 012737 177770 001146 COMPAR: MOV      0-10, ERCOUNT      ;ERROR RETRY COUNTER
1432 010454 052737 000010 001122      BIS      0BIT3, FLAG2      ;DOING COMPARE
1433 010462 013737 001130 001226      MOV      WRDCT, WORK2      ;GET THE WORD COUNT
1434 010470 013737 001116 001150      MOV      VADDR, SAVE      ;SET UP OUTBUFFER POINTER
1435 010476 005037 001174      CLR      SWITCH      ;CLEAR RANDOM PATTERN FLAG
1436 010502 013737 010210 010346      MOV      LOSAV, LONUM1      ;GET RANDOM BASE NOS.
1437 010510 013737 010212 010350      MOV      HISAV, HINUM1
1438 010516 005737 001136      TST      PATNU      ;TEST FOR PATTERN 0
1439 010522 001015      BNE      10      ;NO
1440 010524 005037 001222      CLR      WORK      ;CLEAR COUNTER
1441 010530 062737 000001 001222 20:  ADD      01, WORK      ;INC COUNTER
1442 010536 001003      BNE      30      ;INTERUPT YET?
1443 010540 104054      HLT      IDA:WCIDS      ;TIMED OUT NOT INTERRUPT
1444 010542 000137 001234      JMP      00BEGIN
1445 010546 005737 001176 30:      TST      INTFLG      ;TEST FOR INT
1446 010552 001766      BEQ      20      ;WAIT FOR INT BEFORE COMPARING
1447 010554 000426      BR      CNPLP1      ;CONT
1448 010556 022737 001742 001136 10:  CMP      042, PATNU      ;IS THIS RANDOM PATTERN?
1449 010564 001022      BNE      CNPLP1      ;BRANCH IF YES
1450 010566 005737 00176      CNPLP1: TST      INTFLG      ;INTERUPT YET?
1451 010572 001775      BEQ      CNPLP      ;NO WAIT
1452 010574 005737 001174      TST      SWITCH
1453 010600 001007      BNE      20
1454 010602 004537 010240      JSR      RS, RAND
1455 010606 013701 010346      MOV      LONUM1, GOOD      ;GET EVEN RANDOM WORD
1456 010612 010637 001174      MOV      SP, SWITCH      ;SET RANDOM PATTERN FLAG
1457 010616 000411      BR      WRDCNP
1458 010620 005037 001174 20:      CLR      SWITCH
1459 010624 013701 010350      MOV      HINUM1, GOOD
1460 010630 000404      BR      WRDCNP
1461 010632 013700 001136      CNPLP1: MOV      PATNU, R0
1462 010636 016001 000254      MOV      PAT0(R0), GOOD
1463 010642 160177 170302      WRDCNP: SUB      GOOD, 0SAVE      ;COMPARE DATA
1464 010646 001017      BNE      WDERR      ;WORD IN ERROR
1465 010650 005337 001226      WRDINC: DEC      WORK2      ;DECREMENT THE WORD COUNT
1466 010654 001410      BEQ      ADAN      ;EXIT ROUTINE IF ZERO
1467 010656 062737 000002 001150      ADD      02, SAVE      ;UPDATE PATTERN ADDRESS
1468 010664 022737 000042 001136      CMP      042, PATNU      ;IS THIS RANDOM PATTERN
1469 010672 001735      BEQ      CNPLP      ;BRANCH IF YES
1470 010674 000762      BR      WRDCNP      ;COMPARE NEXT WORD
1471 010676 042737 000010 001122 ADAN:  BIC      0BIT3, FLAG2      ;DONE WITH COMPARE
1472 010704 000205      RTS      R5      ;EXIT THIS ROUTINE

```


1473	010706	005737	001176	WDERR:	TST	INTFLG	;DID INTERRUPT OCCUR YET?
1474	010712	001753			BEQ	WRDCMP	;BRANCH IF NO
1475	010714	032777	000100 170104		BIT	0BIT6,0SWR	;TRY ALL?
1476	010722	001006			BNE	100	;YES
1477	010724	005737	001200		TST	LOPCNT	;FIRST READ ERROR?
1478	010730	001403			BEQ	100	;YES
1479	010732	005777	170074		TST	0RSC81	;ANY ERRORS?
1480	010736	100757			BMI	ADAM	;YES DO NOT COMPARE
1481	010740	060177	170204	100:	ADD	GOOD,0SAVE	
1482	010744	017700	170200		MOV	0SAVE,BAD	;GET GOOD DATA
1483	010750	104436			LOGC		;LOG COMPARE ERROR
1484	010752	032777	001000 170046		BIT	0BIT9,0SWR	;LOOP ON ERROR?
1485	010760	001401			BEQ	110	;NO
1486	010762	005726			TST	(6)+	;YES UPDATE SP
1487	010764	004737	014204	110:	JSR	PC,PRNT	;TYPEOUT?
1488	010770	001007			BNE	30	;NO
1489	010772	104402	010776		TYPE	..+2	;ASCIZ <15><12>"CMP ERR"
1490	011010	104000		30:	HLT		;DATA COMPARE ERROR
1491	011012	004737	014204		JSR	PC,PRNT	;HAD TO DO IT THIS WAY SO
1492	011016	001022			BNE	130	;PROGRAM COULD LOOP ON ERROR
1493	011020	104402	011024		TYPE	..+2	;ASCIZ " ADDR="
1494	011034	005737	001212		TST	MNVA	;IS MEN MONT ON?
1495	011040	001406			BEQ	120	;NO
1496	011042	013746	177776		MOV	PS,-(6)	;GET PS
1497	011046	013746	001150		MOV	SAVE,-(6)	;GET VIRTUAL ADDR
1498	011052	104412			TYPEA		;CONVERT TO PHY AND TYPE
1499	011054	000403			BR	130	;CONT
1500	011056	013746	001150	120:	MOV	SAVE,-(6)	;GET ADDR
1501	011062	104406			TYPEB		;TYPE IT
1502	011064	005037	001154	130:	CLR	BLOCK	;CLEAR THE BLOCK COUNTER
1503	011070	013702	001130		MOV	WRDCT,R2	;GET THE WORD COUNT
1504	011074	005202			INC	R2	;CORRECT FOR DA CALCULATIONS
1505	011076	163702	001226		SUB	WORK2,R2	;DETERMINE DISTANCE OF FAILURE INTO BUFFER
1506	011102	022737	000004 001166	20:	CMP	04,RS04DT	;RS03LAT
1507	011110	001003			BNE	140	;NO
1508	011112	162702	000040		SUB	040,R2	
1509	011116	000410			BR	90	;CONTINUE
1510	011120	005737	001166	140:	TST	RS04DT	;RS04?
1511	011124	001403			BEQ	70	;NO
1512	011126	162702	000200		SUB	0200,R2	;RS03
1513	011132	000402			BR	90	;CONT
1514	011134	162702	000100	70:	SUB	0100,R2	
1515	011140	100403		90:	BNI	00	
1516	011142	005237	001154		INC	BLOCK	;UPDATE BLOCK COUNT FOR EACH 400 WORDS
1517	011146	000755			BR	20	

1518	011150	022737	000004	001166	001	CMP	#4,R804DT	;R804LA
1519	011156	001003				BNE	150	;NO
1520	011160	062702	000040			ADD	#40,R2	;RESTORE POSITIVE :
1521	011164	000410				BR	60	;CONTINUE
1522	011166	005737	001166		150:	TST	R804DT	;R804?
1523	011172	001403				BEQ	40	;NO
1524	011174	062702	000200			ADD	#200,R2	;R804
1525	011200	000402				BR	60	;CONT
1526	011202	062702	000100		40:	ADD	#100,R2	;RESTORE POSITIVE NUMBER
1527	011206	013737	001134	001224	60:	MOV	DMA,WORK1	;GET HEAD AND SECTOR ADDRESS
1528	011214	063737	001154	001224	50:	ADD	BLOCK,WORK1	
1529	011222	004737	014204			JSR	PC,PRNT	;TYPEOUT?
1530	011226	001014				BNE	10	;NO
1531	011230	104402	011234			TYPE	.,+2	;ASCIZ "DA"
1532	011242	013746	001224			MOV	WORK1,-(6)	;PUT WORK1 ON STACK
1533	011246	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1534	011250	104402	011254			TYPE	.,+2	;ASCIZ <15><12>
1535	011260	032777	000020	167540	10:	BIT	#BIT4,05WP	;RETRY?
1536	011266	001405				BEQ	CLEAR	;NO
1537	011270	005237	001146			INC	ERCOUNT	;UPDATE ERROR COUNTER
1538	011274	001402				BEQ	CLEAR	
1539	011276	000137	010650			JNP	WRDINC	
1540	011302	032737	000004	001126	CLEAR:	BIT	#BIT2,FLAG	;XFER TEST?
1541	011310	001404				BEQ	30	;NO
1542	011312	032737	010000	001126		BIT	#BIT12,FLAG	;READY?
1543	011320	001412				BEQ	20	;NO
1544	011322	013700	001116		30:	MOV	VADDR,R0	;GET STARTING ADDR OF BUFFER
1545	011326	013701	001130			MOV	WRDCT,R1	;NOW
1546	011332	005020			10:	CLR	(R0)+	;CLEAR BUFFER
1547	011334	005301				DEC	R1	;COUNT LOCATIONS
1548	011336	001375				BNE	10	;WAIT TILL DONE
1549	011340	042737	000010	001122		BIC	#BIT3,FLAG2	;DONE WITH COMPARE
1550	011346	000205			20:	RTS	R5	;NOW GET OUT
1551								
1552	011350	013737	001072	017416	APORT:	MOV	SAVAST,OUTBUF	;SET STARTING ADDR FOR OUTBUF
1553	011356	013737	001072	001116		MOV	SAVAST,VADDR	;SAVE OUTBUF ADDR
1554	011364	005737	001212			TST	NNAVA	;MEN NGMT?
1555	011370	001411				BEQ	EXTT	;NO
1556	011372	013702	001100			MOV	SAVNGA,R2	;SET UP MEN NGMT
1557	011376	004737	012020		NNSET:	JSR	PC,STNN2	;SETUP MEN NGMT
1558	011402	010237	001116			MOV	R2,VADDR	
1559	011406	013737	001120	017416		MOV	PHADDR,OUTBUF	
1560	011414	000207			EXTT:	RTS	PC	
1561								
1562	011416	013737	001112	001130	BPORT:	MOV	WDCTB,WRDCT	;GET WC FOR B PORT
1563	011424	013737	001076	017416		MOV	SAVCPU,OUTBUF	
1564	011432	013737	001076	001116		MOV	SAVCPU,VADDR	
1565	011440	005737	001212			TST	NNAVA	;MEN NGMT AVAILABLE?
1566	011444	001763				BEQ	EXTT	;NO
1567	011446	013702	001104			MOV	SAVNGC,R2	
1568	011452	000751				BR	NNSET	


```

1569                                     ;TYPE CAN NOT WRITE BLOCK
1570
1571 011454 004737 014204      WTN01  JSR      PC,PRNT      ;TYPEOUT?
1572 011460 001001              BNE      10              ;NO
1573 011462 000000              HALT                      ;HALT CANT WRITE BLOCK
1574 011464 005037 001200      101    CLR      LOPCNT      ;CLEAR ERR COUNTER

1575 011470 000207              RTS      PC

1576
1577                                     ;ROUTINE TO SET UP STARTING ADDRESS FOR ALL PORTS
1578                                     ;AND TO CREATE WORD COUNT MAX= 20K
1579
1580 011472 013702 001070      EXTHEN: NOV      STANEN,R2      ;GET BANK 0
1581 011476 005702              TST      R2              ;DID HE TYPE 0?
1582 011500 001001              BNE      30              ;NO
1583 011502 005302              INC      R2              ;YES MAKE 1
1584 011504 005737 001212      301    TST      MMVA          ;BRANCH IF MMV MGMT AVAILABLE
1585 011510 001021              BNE      10
1586 011512 000241              CLC
1587 011514 004737 011736      JSR      PC,RRR2
1588 011520 010237 001072      NOV      R2,SAVAST      ;SAVE A STARTING ADDR
1589 011524 032737 000100 001126      BIT      0BIT6,FLAG      ;IS THERE A B PORT?
1590 011532 001430              BEQ      20              ;NO
1591 011534 013702 001074      NOV      STDCON,R2      ;YES GET STARTING
1592 011540 000241              CLC
1593 011542 004737 011736      JSR      PC,RRR2
1594 011546 010237 001076      NOV      R2,SAVCPU      ;SAVE IT
1595 011552 000420              BR       20              ;GET WC
1596 011554 000302      101    SWAB      R2
1597 011556 006002              ROR      R2
1598 011560 010237 001100      NOV      R2,SAVNGA      ;SAVE ADDR FOR A PORT
1599 011564 032737 000100 001126      BIT      0BIT6,FLAG      ;IS THERE B PORT?
1600 011572 001410              BEQ      20              ;NO
1601 011574 013702 001074      NOV      STDCON,R2
1602 011600 000302              SWAB      R2
1603 011602 006002              ROR      R2
1604 011604 010237 001102      NOV      R2,SAVNGB      ;SAVE B STARTING ADDR
1605 011610 010237 001104      NOV      R2,SAVNGC      ;SAVE CPU STARTING ADDR
1606 011614 013702 001106      NOV      SIZEAP,R2      ;GET 4K BLOCK COUNT
1607 011620 022737 000007 001106      CMP      07,SIZEAP      ;IS IT GREATER THEN 20K?
1608 011626 101411              BLOS      40              ;YES MAKE IT 20K

```


1609	011630	000241		88:	CLC		
1610	011632	006002			ROR	R2	;NO CONVERT TO WC
1611	011634	004737	011736		JSR	PC,RRR2	
1612	011640	042702	000077		BIC	077,R2	;CLEAR BLOCK COUNT
1613	011644	010237	001144		MOV	R2,SWRDCT	;SAVE -A- PORT WC
1614	011650	000403			BR	50	;CONT
1615	011652	012737	060000	001144	40:	MOV	060000,SWRDCT
1616	011660	032737	000100	001126	50:	BIT	0BIT6,FLAG
1617	011666	001422			BEG	70	;IS THERE B PORT?
1618	011670	013702	001110		MOV	SIZEBP,R2	;NO GET OUT
1619	011674	022737	000007	001110		CMP	07,SIZEBP
1620	011702	101411			BLOS	60	;GET B 4K COUNT
1621	011704	000241		90:	CLC		;IS IT GREATER THEN 20K?
1622	011706	006002			ROR	R2	;YES MAKE 20K
1623	011710	004737	011736		JSR	PC,RRR2	;NO CONVERT TO WC
1624	011714	042702	000077		BIC	077,R2	;CLEAR SECTOR COUNT
1625	011720	010237	001112		MOV	R2,WDCTB	;SAVE WC FOR -B- PORT
1626	011724	000403			BR	70	;GET OUT
1627	011726	012737	060000	001112	60:	MOV	060000,WDCTB
1628	011734	000207		70:	RTS	PC	;MADE 20K WD
1629							
1630	011736	006002		RRR2:	ROR	R2	
1631	011740	006002			ROR	R2	
1632	011742	006002			ROR	R2	
1633	011744	006002			ROR	R2	
1634	011746	000207			RTS	PC	
1635							
1636	011750	032777	000200	167050	WATT:	BIT	0BIT7,0SWR
1637	011756	001003			BNE	10	;WAIT IN BACKGROUND?
1638	011760	004737	012312		JSR	PC,XWAIT	;NO
1639	011764	000401			BR	20	;YES
1640	011766	000001		10:	WAIT		;CONT
1641	011770	000207		20:	RTS	PC	
1642							
1643	011772	004737	014204		TYPREC:	JSR	PC,PRNT
1644	011776	001007			BNE	10	;TYPEOUT?
1645	012000	104402	000616		TYPE	,RECOV	;NO
1646	012004	013746	001200		MOV	LOPCNT,-(6)	;GET COUNT
1647	012010	104406			TYPES		;TYPE IT
1648	012012	104402	000636		TYPE	,CRLF	
1649	012016	000207		10:	RTS	PC	

1650								
1651	012020	005737	001212		STNM2:	TST	MNAVA	;MEN NGMT?
1652	012024	001002				BNE	30	;YES
1653	012026	000137	012304			JMP	MDON	;GET OUT
1654	012032	005037	172340		30:	CLR	00KIPAR0	
1655	012036	010237	001150			NOV	R2,SAVE	;SAVE R2
1656	012042	010237	172342			NOV	R2,00KIPAR1	
1657	012046	006302				ASL	R2	;CALCULATE PHYSICAL ADDR
1658	012050	006302				ASL	R2	
1659	012052	006302				ASL	R2	
1660	012054	006302				ASL	R2	
1661	012056	006302				ASL	R2	;THIS BIT IS A17
1662	012060	042737	000040	001122		BIC	0BITS,FLAG2	;CLEAR A17
1663	012066	103003				BCC	10	;SET A17
1664	012070	052737	000040	001122		BIS	0BITS,FLAG2	;SET BIT 5 FOR A17
1665	012076	042737	000020	001122	10:	BIC	0BIT4,FLAG2	;CLEAR A16 FLAG
1666	012104	006302				ASL	R2	;GET A16 BIT
1667	012106	103003				BCC	20	;CLEAR A16
1668	012110	052737	000020	001122		BIS	0BIT4,FLAG2	;SET FLAG FOR A16
1669	012116	010237	001120		20:	NOV	R2,PHADDR	;GET PHYSICAL ADDR
1670	012122	013702	001150			NOV	SAVE,R2	;SET UP MEN NGMT
1671	012126	062702	000200			ADD	0200,R2	
1672	012132	010237	172344			NOV	R2,00KIPAR2	
1673	012136	062702	000200			ADD	0200,R2	
1674	012142	010237	172346			NOV	R2,00KIPAR3	
1675	012146	062702	000200			ADD	0200,R2	
1676	012152	010237	172350			NOV	R2,00KIPAR4	
1677	012156	062702	000200			ADD	0200,R2	
1678	012162	010237	172352			NOV	R2,00KIPAR5	
1679	012166	062702	000200			ADD	0200,R2	
1680	012172	010237	172354			NOV	R2,00KIPAR6	
1681	012176	012737	077406	172300		NOV	0200+256,-400+UP+RN,00KIPDR0	;SET KIPDR0=RN UP 200 BLOCKS
1682	012204	012737	077406	172302		NOV	0200+256,-400+UP+RN,00KIPDR1	;SET KIPDR1=RN UP 200 BLOCKS
1683	012212	012737	077406	172304		NOV	0200+256,-400+UP+RN,00KIPDR2	;SET KIPDR2=RN UP 200 BLOCKS
1684	012220	012737	077406	172306		NOV	0200+256,-400+UP+RN,00KIPDR3	;SET KIPDR3=RN UP 200 BLOCKS
1685	012226	012737	077406	172310		NOV	0200+256,-400+UP+RN,00KIPDR4	;SET KIPDR4=RN UP 200 BLOCKS
1686	012234	012737	077406	172312		NOV	0200+256,-400+UP+RN,00KIPDR5	;SET KIPDR5=RN UP 200 BLOCKS
1687	012242	012737	077406	172314		NOV	0200+256,-400+UP+RN,00KIPDR6	;SET KIPDR6=RN UP 200 BLOCKS
1688	012250	012737	077406	172316		NOV	0200+256,-400+UP+RN,00KIPDR7	;SET KIPDR7=RN UP 200 BLOCKS
1689	012256	012737	007600	172356		NOV	07600,00KIPAR7	
1690	012264	012702	020000			NOV	020000,R2	
1691	012270	012737	012306	000250		NOV	0MNAUTO,00HNVEC	
1692	012276	012737	000001	177572		NOV	01,00SR0	;TURN ON MEN NGMT
1693	012304	000207			MDON:	RTS	PC	
1694								
1695								
1696	012306	000000						
1697	012310	000002						

;MEMORY MANAGEMENT ABORT ROUTINE FOR WRITE UP
MNAUTO: HALT
RTI

;MEMORY MANAGEMENT TRAP
CAUSED THE ABORT


```

1698                                     ;BACKGROUND TEST FOR INTERRUPTS
1699
1700 012312 052737 002000 001122 XWAIT: BIS 0BIT10,FLAG2 ;WAITING IN BACKGROUND TEST
1701 012320 012737 070000 012426      MOV 070000,NPRCNT ;SETUP TIMEOUT COUNTER
1702 012326 012701 012431      MOV 0NPR1+1,R1 ;SETUP WAIT LOOP
1703 012332 112711 000200      MOVB 0200,(R1)
1704 012336                                     20:
1705 012336 105421      NEGB (R1)+
1706 012340 105441      NEGB -(R1)
1707 012342 105421      NEGB (R1)+
1708 012344 105441      NEGB -(R1)
1709 012346 105421      NEGB (R1)+
1710 012350 105441      NEGB -(R1)
1711 012352 105421      NEGB (R1)+
1712 012354 105441      NEGB -(R1)
1713 012356 105421      NEGB (R1)+
1714 012360 105441      NEGB -(R1)
1715 012362 105421      NEGB (R1)+
1716 012364 105441      NEGB -(R1)
1717 012366 105421      NEGB (R1)+
1718 012370 105441      NEGB -(R1)
1719 012372 105421      NEGB (R1)+
1720 012374 105441      NEGB -(R1)
1721 012376 102401      BVS 10
1722 012400 000000      HALT
1723 012402 005337 012426 10: DEC NPRCNT ;ARITHMETIC OPERATION FAILED RUN DIAG
1724 012406 001353      SNE 20
1725 012410 104054      HLT 10A1NC1DS ;TIMED OUT NO INTERRUPT
1726 012412 000137 001234      JMP 00BEGIN
1727 012416 042737 002000 001122 NPRRET: BIS 0BIT10,FLAG2 ;CLEAR BKGROUND FLG
1728 012424 000207      RTS PC
1729 012426 000000      NPRCNT: 0
1730 012430 000000      NPR1: 0
1731                                     ;CLEAR ERROR TABLE
1732
1733 012432 012704 000020      ERRCL: MOV 020,R4 ;CLEAR
1734 012436 012703 017052      MOV 0ERTAB,R3 ;ERROR
1735 012442 005023 10: CLR (R3)+ ;TABLE
1736 012444 005304      DEC R4 ;DONE YET?
1737 012446 001375      SNE 10 ;NO
1738 012450 005037 001004      CLR PCNT ;CLEAR
1739 012454 005037 001006      CLR PCNT+2 ;PASS COUNT
1740 012460 005037 001124      CLR DROP ;CLEAR ALL DROPPED DRIVES
1741 012464 000205      RTS R5 ;RETURN

```


1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780

012466	012706	000500	
012472	104402	000510	
012476	104420		
012500	012637	001160	
012504	004737	006774	
012510	004737	006840	
012514	104426		
012516	004737	013372	
012522	005037	001134	
012526	012737	012750	000024
012534	012737	000340	000026
012542	012737	000161	001172
012550	104416		
012552	004737	011750	
012556	032737	001000	001126
012564	001406		
012566	104006		
012570	012777	177777	166292
012576	005077	166244	
012602	004737	007050	
012606	000755		
012610	000744		

;RH11 POWER FAIL TEST 01
;THE STARTING ADDRESS FOR THE WRITE POWER FAIL TEST IS 244. THE PROGRAM
;WRITE THE COMPLETE DISK WITH A 125252 PATTERN. THE PROGRAM WILL THEN
;TELL OPERATOR TO POWER DOWN. UNTIL THE POWER FAIL, THE PROGRAM WILL
;CONTINUE WRITING THE SAME PATTERN ON THE DISK.
;WHEN POWER FAIL OCCURS THE TRANSFER IS ABORTED
;AND THE PROGRAM HALTS. THE OPERATOR SHOULD
;NOW TURN POWER BACK ON. THE PROGRAM RESTARTS AND CHECKS FOR WRITE ERRORS.
;ONLY ONE ERROR IS ACCEPTABLE. THAT ERROR MAY BE AN OPI (BIT13 RSR)OR A DCK
;(BIT 15 RSR). IF THESE ARE THE ONLY ERRORS THAT OCCUR, THE DRIVE IS OK.
;IF NO ERRORS OCCUR, THE PROGRAM WILL TYPE OUT "OK".
;THE PROGRAM WILL THEN TELL YOU WHEN TO POWER DOWN AGAIN

;***ONLY ONE ERROR IS CONSIDERED ACCEPTABLE***
;NOTE: ALL DRIVES ON THE SYSTEM SHOULD BE POWERED OFF EXCEPT
;THE DRIVE UNDER TEST. *****

PFT1:	MOV	0500,SP	;SET UP STACK
	TYPE	,LOADSW	
	RDOCT		
	MOV	(SP)+,UNNUM	
20:	JSR	PC,VECTR	;SETUP INT VECTOR
	JSR	PC,FNDTYP	;TST FOR RS03 OR 04
PFWATT:	CLRDV		;CLEAR ALL REG
	JSR	PC,PONFAL	;WRITE 125252 ON DISK
PFWAT:	CLR	DNA	
	MOV	0DOWN,24	;SET UP POWER FAIL VEC.
	MOV	0340,26	
HYBYWR:	MOV	0161,CND	;WRITE WITH I/E
	DKCHD		;DO IT
	JSR	PC,WATT	;WAIT FOR INTERRUPT
30:	BIT	0BIT9,FLAG	;ANY ERRORS?
	BEG	10	;NO
	HLT	IDAIDS	
	MOV	0-1,0RSAS	;CLEAR ALL
	CLR	0RSR	;ERRORS
10:	JSR	PC,DISBUF	;SET UP NEW DISK BUFFER
	BR	HYBYWR	
	BR	PFWAT	

1781	012612	012737	012620	001152	UPCHK:	MOV	010,HRDR	;RETURN HERE IF WRONG DRIVE INTERRUPTS
1782	012620	005037	001134		10:	CLR	DMA	
1783	012624	104426				CLRDV		;INIT DRIVE
1784	012626	013737	001066	177776	CHKDAT:	MOV	PRIORITY,PS	
1785	012634	012737	000151	001172		MOV	0151,CMD	;WRITECHECK WITH I/E
1786	012642	104416				DKCMD		;DO IT
1787	012644	013737	001066	177776		MOV	PRIORITY,PS	
1788	012652	004737	011750			JSR	PC,WATT	;WAIT FOR INTERRUPT
1789	012656	032737	001000	001126	30:	BIT	0BIT9,FLAG	;ANY ERRORS?
1790	012664	001411				BEG	10	;NO
1791	012666	104006				HLT	IDBIDA	
1792	012670	052737	100000	001122		BIS	0BIT15,FLAG2	;SET ERROR FLAG
1793	012676	005077	166144			CLR	ORSER	;CLEAR ALL
1794	012702	012777	177777	166140		MOV	0-1,ORSA8	;ERRORS
1795	012710	004737	007050		10:	JSR	PC,DISBUF	;SET UP NEW DISK BUFFER
1796	012714	000744				BR	CHKDAT	
1797	012716	005737	001122			TST	FLAG2	;ANY ERRORS?
1798	012722	100405				BNI	20	;YES
1799	012724	104402	012730			TYPE	,,+2	;,ASCII <15><12>"OK"
1800	012736	042737	100000	001122	20:	BIC	0BIT15,FLAG2	;CLEAR ERROR FLAG
1801	012744	000137	012514			JMP	PFWATT	;GO WAIT FOR ANOTHER
1802								;POWER FAIL
1803								
1804								
1805								;POWER DOWN ROUTINE - ABORT DISK AND HALT
1806	012750	012737	012760	000024	DOWN:	MOV	0UPP,24	;SET POWER FAIL VECTOR
1807	012756	000000				HALT		
1808								
1809	012760	012737	012750	000024	UPP:	MOV	0DOWN,24	
1810	012766	012706	000500			MOV	0500,0P	
1811	012772	013777	001160	166034		MOV	UNNUM,ORSC62	;GET UNIT 0
1812	013000	032777	000200	166036	10:	BIT	0BIT7,ORSD8	;WAIT FOR DRIVE READY
1813	013006	001774				BEG	10	
1814	013010	000137	012612			JMP	UPCHK	;GO CHECK DISK

1015
1016
1017
1018
1019
1020

;POWER FAIL TEST #2
;THIS TEST WILL TEST THE SAME DRIVE THAT WAS TESTED IN THE 1ST POWER FAIL TEST
;THE PROGRAM WILL WRITE THE COMPLETE DISK WITH A 125252 PATTERN AND WILL
;THEN TELL THE OPERATOR TO POWER DOWN THE PROCESSOR.
;THE PROGRAM WILL THEN WRITE CHECK THE DISK WHILE WAITING FOR A POWER FAIL.
;WHEN THE POWER FAIL OCCURS, THE WRITE CHECKING IS ABORTED AND

1021
1022
1023
1024
1025
1026
1027
1028
1029
1030

;THE PROCESSOR WILL HALT.
;THE OPERATOR SHOULD THEN TURN POWER BACK ON, THE PROGRAM WILL
;START WRITE CHECKING THE DISK AGAIN
;***NO ERRORS SHOULD OCCUR***
;THE PROGRAM WILL TYPE OUT "OK" IF NO ERRORS OCCUR.
;THE PROGRAM WILL THEN TELL YOU TO POWER DOWN.
;DO NOT POWER OFF THE PROCESSOR AGAIN UNTIL THE PROGRAM TELLS YOU SO.
;NOTE: ALL DRIVES ON THE SYSTEM SHOULD BE POWERED OFF
;EXCEPT THE ONE UNDER TEST *****

1031 013014 012706 000500
1032 013020 042737 001000 001122
1033 013026 012737 013050 001152
1034 013034 104426
1035 013036 004737 006774
1036 013042 004737 013272
1037 013046 000401
1038 013050 104426
1039 013052 005037 001134
1040 013056 012737 013222 000024
1041 013064 012737 000340 000026
1042 013072 013737 001066 177776
1043 013100 012737 000151 001172
1044 013106 104416
1045 013110 004737 011750
1046 013114 032737 001000 001126 301
1047 013122 001411
1048 013124 104002
1049 013126 002737 100000 001122
1050 013134 005077 165706
1051 013140 012777 177777 165702
1052 013146 004737 007050 101
1053 013152 000747
1054 013154 032737 001000 001122
1055 013162 001733
1056 013164 005737 001122
1057 013170 100405
1058 013172 104402 013176
1059 013204 042737 100000 001122 201
1060 013212 042737 001000 001122
1061 013220 000710 401

PFT2: NOV 0500,SP ;SET UP STACK
BIC 0BIT9,FLAG2 ;CLEAR POWER FAIL
NOV 0PWRFL1,WRDR ;RETURN HERE IF WRONG DRIVE INT.
CLRDV ;INIT DRIVE
JSR PC,VECTRR ;SETUP INT VECTOR
PWRFL2: JSR PC,PONFAL ;WRITE 125252 ON DISK
BR PWRFL ;WRITE CHECK
PWRFL1: CLRDV ;INIT DRIVE
PWRFL: CLR DMA
NOV 0PWRDN,24 ;SET UP POWER FAIL VEC.
NOV 0340,26
CHKDSK: NOV PRIORITY,PS ;ENABLE I/E
NOV 0151,CND ;WRITE CHECK WITH I/E
DKCND ;DO IT
JSR PC,WATT ;WAIT FOR INTERRUPT
BIT 0BIT9,FLAG ;ANY ERRORS?
BEQ 10 ;NO
HLT 100 ;YES
BIS 0BIT15,FLAG2 ;SET ERROR FLAG
CLR ORSR ;CLEAR ALL
NOV 0-1,ORSR ;ERRORS
JSR PC,DIBUF ;CHECK NEXT BUFFER
BR CHKDSK
BIT 0BIT9,FLAG2 ;DID POWER FAIL?
BEQ PWRFL ;NO
TST FLAG2 ;ANY ERRORS?
BNI 20 ;YES
TYPE ,,+2 ;,ASCII <15><12>"OK"
BIC 0BIT15,FLAG2 ;CLEAR ERRORS
BIC 0BIT9,FLAG2 ;CLEAR POWER FAIL FLAG
BR PWRFL2


```

1862                                     ;ROUTINE TO ABORT DISK DURING POWER FAIL
1863
1864 013222 012737 013232 000024 PWRDN: MOV      $PWRUP,24      ;SET UP RESTART
1865 013230 000000                                HALT
1866
1867 013232 012737 013222 000024 PWRUP: MOV      $PWRDN,24      ;RESET POWER FAIL VECTOR
1868 013240 012706 000500                                MOV      $500,$P
1869 013244 013777 001160 165562                                MOV      UNNUM,$RSC62    ;GET UNIT 0
1870 013252 052737 001000 001122                                BIS      $BIT9,FLAG2    ;SET POWER FAIL BIT
1871 013260 032777 000200 165556 10: BIT      $B7,T7,$RSD8    ;WAITING FOR
1872 013266 001774                                BEQ      10              ;DRIVE READY
1873 013270 000667                                BR        PWRF1         ;GO CHECK DISK
1874
1875
1876                                     ;ROUTINE TO WRITE THE COMPLETE DISK
1877                                     ;WITH 128282 PATTERN
1878                                     ;WRITE CHECK AND REPORT ERRORS IF THEY OCCUR
1879                                     ;REPORT "OK" AT COMPLETION
1880
1881 013272 012737 000020 001136 POWFAL: MOV      $20,PATNU      ;SET UP PATTERN
1882 013300 042737 000004 001126                                BIC      $BIT2,FLAG    ;CLEAR XFER MODE FLAG
1883 013306 052737 010000 001122                                BIS      $BIT12,FLAG2
1884 013314 005037 001134                                CLR      DMA
1885 013320 012737 020000 017416                                MOV      $20000,OUTBUF  ;GET STARTING ADDR FOR BUF
1886 013326 012737 020000 001116                                MOV      $20000,VADDR
1887 013334 012737 010000 001144                                MOV      $10000,$WRDCT  ;SETUP WORD COUNT
1888 013342 013737 001144 001130                                MOV      $WRDCT,$WRDCT
1889 013350 005037 001114                                CLR      AOB1
1890 013354 013737 017416 001140                                MOV      OUTBUF,BUF     ;A PORT ONLY
1891 013362 004537 007716                                JSR      R5,$ASEL        ;SET UP CURRENT ADDRESS
1892 013366 012737 000161 001172 WRDWN: MOV      $161,CMD      ;GENERATE DATA BUFFER
1893 013374 104416                                DKCND
1894 013376 004737 011750                                JSR
1895 013402 012737 000151 001172 20: MOV      $151,CMD      ;WRITE WITH I/E
1896 013410 104416                                DKCND
1897 013412 004737 011750                                JSR
1898 013416 032737 001000 001126 40: BIT      $BIT9,FLAG    ;DO IT
1899 013424 001402                                BEQ      10              ;WAIT FOR INTERRUPT
1900 013426 104006                                HLT      10              ;WRITECHECK I/E
1901 013430 000000                                HALT
1902 013432 004737 007050 10: JSR      PC,$DISBUF    ;DO IT
1903 013436 000753                                BR
1904 013440 104402 000743                                TYPE     ;ANY ERRORS?
1905 013444 000207                                RTS      PC              ;NO
                                ;YES
                                ;CAN NOT WRITE WITHOUT ERROR
                                ;SET UP NEW DISK BUFFER
                                ;WRITE NEW BUFFER

```


1906	013446	032777	000010	165352	OUT:	BIT	0BIT3,0SHR	;TYPEOUT ERROR COUNT?
1907	013454	001526				BEQ	10	;NO
1908	013456	005004				CLR	R4	;CLEAR UNIT 0
1909	013460	005003				CLR	R3	
1910	013462	053737	001124	001162		BIS	DROP,UNITSV	;RESTORE ALL DRIVES
1911	013470	013737	001162	001222		MOV	UNITSV,WORK	;GET UNITS ON SYSTEM
1912	013476	012705	000401			MOV	0401,R5	;SETUP TEST FOR UNITS
1913	013502	030537	001222		40:	BIT	R5,WORK	;IS THIS UNIT ON SYS
1914	013506	001006				BNE	20	;YES
1915	013510	005204			50:	INC	R4	;INC UNIT 0
1916	013512	010403				MOV	R4,R3	;SAVE UNIT 0
1917	013514	000241				CLC		
1918	013516	006105				ROL	R5	;GET NEXT DRIVE
1919	013520	103501				BCS	30	;DONE
1920	013522	000767				BR	40	;FIND NEXT DRIVE
1921	013524	104402	000510		20:	TYPE	,LOADSW	
1922	013530	010446				MOV	R4,-(6)	;PUT R4 ON STACK
1923	013532	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1924	013534	004737	014012			JSR	PC,GETERR	;GET ERROR COUNT
1925	013540	010304				MOV	R3,R4	;RESTORE UNIT 0
1926	013542	104402	013546			TYPE	,,+2	;ASCIZ <15><12>
1927	013552	104402	000564			TYPE	,WRERR	
1928	013556	104402	013562			TYPE	,,+2	;ASCIZ "S "
1929	013566	013746	001202			MOV	WRITER,-(6)	;PUT WRITER ON STACK
1930	013572	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1931	013574	104402	013600			TYPE	,,+2	;ASCIZ <15><12>
1932	013604	104402	000607			TYPE	,RDERR	
1933	013610	104402	013614			TYPE	,,+2	;ASCIZ "S "
1934	013620	013746	001206			MOV	READER,-(6)	;PUT READER ON STACK
1935	013624	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1936	013626	104402	013632			TYPE	,,+2	;ASCIZ <15><12>
1937	013636	104402	000574			TYPE	,WCKERR	
1938	013642	104402	013646			TYPE	,,+2	;ASCIZ "S "
1939	013652	013746	001204			MOV	WCERR,-(6)	;PUT WCERR ON STACK
1940	013656	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1941	013660	104402	013664			TYPE	,,+2	;ASCIZ <15><12>"COMPARE ERRS "
1942	013704	013746	001210			MOV	CONERR,-(6)	;PUT CONERR ON STACK
1943	013710	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1944	013712	104402	013716			TYPE	,,+2	;ASCIZ <15><12>
1945	013722	000672				BR	50	;GET NEXT DRIVE
1946	013724	043737	001124	001162	30:	BIC	DROP,UNITSV	;REDROP DRIVES
1947	013732	062706	000002		10:	ADD	02,SP	;RESTORE SP DUE TO JMP EXIT FROM JSR ROUTINE
1948	013736	005137	001114			CON	ASB1	;SET A OR B PORT FLAG
1949	013742	032777	000040	165056		BIT	0BITS,0SHR	;TYPEOUT PASS COUNT?
1950	013750	001035				BNE	DONE	;NO
1951	013752	104402	013756			TYPE	,,+2	;ASCIZ <15><12>"END PASS "
1952	013772	013746	001006			MOV	PCNT+2,-(6)	;PUT PCNT+2 ON STACK
1953	013776	104406				TYPE		;TYPE STACK IN OCTAL - SUPRESS
1954	014000	104402	014004			TYPE	,,+2	;ASCIZ <15><12>
1955	014010	000415				BR	DONE	

1956	014012	006304			GETERR:	ASL	R4		;GET LOC IN
1957	014014	006304				ASL	R4		;ERR TABLE
1958	014016	0062704	017052			ADD	ERTAB,R4		
1959	014022	112437	001202			MOVB	(R4)+,WRITER		;GET WRITE ERRS
1960	014026	112437	001206			MOVB	(R4)+,READER		;GET READ ERRS
1961	014032	112437	001204			MOVB	(R4)+,WCERR		;GET WRITE CK ERRS
1962	014036	112437	001210			MOVB	(R4)+,COMERR		;GET COMPARE ERRS
1963	014042	000207				RTS	PC		
1964									
1965					.SBTTL			SDONE - BELL AND SCOPE ROUTINE	
1966									
1967	014044	104400			DONE:	SCOPE			;TERMINATIONS SCOPE FOR LOOPING
1968	014046	062737	000001	001006		ADD	01,PCNT+2		;ADD 1 TO THE PASS COUNT
1969	014054	005537	001004			ADC	PCNT		;MAKE IT DOUBLE PREC.
1970	014060	013700	000042		40:	MOV	0042,R0		;GET MONITOR ADDRESS
1971	014064	001405				BEG	SEND1		;IF NONE
1972	014066	000005				RESET			
1973	014070	004710			SENDAD:	JSR	7,(0)		;GO TO MONITOR
1974	014072	000240	000240	000240			240,240,240		;SAVE ROOM FOR ACT11
1975	014100	000137	003224		SEND1:	JMP	ADTST		;RETURN
1976									
1977	014104	000000			.TBIT:	0			;T BIT FLAG
1978									
1979	014106	012702	000001		.LOGW:	MOV	01,R2		;LOG WRITE ERR
1980	014112	005003			CLIND:	CLR	R3		;CLEAR INDEX FOR TABLE
1981	014114	000413				BR	ADDRR		
1982									
1983	014116	012702	000400		.LOGR:	MOV	0400,R2		;LOG WRITE ERR
1984	014122	000773				BR	CLIND		
1985									
1986	014124	012702	000001		.LOGWC:	MOV	01,R2		;LOG WRITE CK ERR
1987	014130	012703	000002		SETIND:	MOV	02,R3		;SET INDEX FOR NEXT WD
1988	014134	000403				BR	ADDRR		
1989									
1990	014136	012702	000400		.LOGC:	MOV	0400,R2		;LOG COMPARE ERR
1991	014142	000772				BR	SETIND		
1992									
1993	014144	005737	001200		ADDRR:	TST	LOPCNT		;1ST TIME ERROR?
1994	014150	001014				BNE	10		;NO DO NOT COUNT IT
1995	014152	013704	001160			MOV	UNNUM,R4		;GET UNIT 0
1996	014156	006304				ASL	R4		;GET
1997	014160	006304				ASL	R4		;POSITION IN
1998	014162	006304				ADD	R3,R4		;ERR TABLE
1999	014164	006264	017052			ADD	R2,ERTAB(R4)		;TO ADD ERROR
2000	014170	004737	014204			JSR	PC,PRNT		;TYPEOUT?
2001	014174	001402				BEG	10		;YES
2002	014176	004737	014772			JSR	PC,DRP		;SHOULD I DROP DRIVE?
2003	014202	000002			10:	RTI			
2004									
2005	014204	032777	020000	164614	PRNT:	BIT	0BIT13,06WR		;INHIBIT TYPEOUT?
2006	014212	000207				RTS	PC		

2007	014214	052737	000004	001122	RSREG1	BIS	%BIT2,FLAG2	;SET ERROR FLAG
2008	014222	005737	016040			TST	.HLTCT	;SHOULD WE TYPE GOOD AND BAD
2009	014226	001017				BNE	00	;NO
2010	014230	104402	014234			TYPE	..+2	;ASCIZ "BAD"
2011	014242	010046				MOV	BAD,-(6)	;PUT BAD ON STACK
2012	014244	104404				TYPEO		;TYPE STACK IN OCTAL
2013	014246	104402	014252			TYPE	..+2	;ASCIZ "GOOD"
2014	014262	010146				MOV	GOOD,-(6)	;PUT GOOD ON STACK
2015	014264	104404				TYPEO		;TYPE STACK IN OCTAL
2016	014266			001				
2017	014266	104402	014272			TYPE	..+2	;ASCIZ "CS1"
2018	014300	017746	164526			MOV	ORSCS1,-(6)	;PUT ORSCS1 ON STACK
2019	014304	104404				TYPEO		;TYPE STACK IN OCTAL
2020	014306			101				
2021	014306	104402	014312			TYPE	..+2	;ASCIZ "ER"
2022	014320	017746	164522			MOV	ORSER,-(6)	;PUT ORSER ON STACK
2023	014324	104404				TYPEO		;TYPE STACK IN OCTAL
2024	014326			201				
2025	014326	104402	014332			TYPE	..+2	;ASCIZ "CS2"
2026	014340	017746	164470			MOV	ORSCS2,-(6)	;PUT ORSCS2 ON STACK
2027	014344	104404				TYPEO		;TYPE STACK IN OCTAL
2028	014346	032737	000200	016040		BIS	%200,.HLTCT	;PRINT SECOND SET ?
2029	014354	001112				BNE	SEEC	;YES
2030	014356	032737	000100	016040		BIS	%AS,.HLTCT	;PRINT ER ?
2031	014364	001410				BEO	30	;NO
2032	014366	104402	014372			TYPE	..+2	;ASCIZ "AS"
2033	014400	017746	164444			MOV	ORAS,-(6)	;PUT ORAS ON STACK
2034	014404	104404				TYPEO		;TYPE STACK IN OCTAL
2035	014406	032737	000020	016040	301	BIS	%DA,.HLTCT	;PRINT BUS ADDRESS
2036	014414	001410				BEO	40	;NO
2037	014416	104402	014422			TYPE	..+2	;ASCIZ "DA"
2038	014430	017746	164404			MOV	ORSDA,-(6)	;PUT ORSDA ON STACK
2039	014434	104404				TYPEO		;TYPE STACK IN OCTAL
2040	014436	032737	000004	016040	401	BIS	%DA,.HLTCT	;PRINT DA ?
2041	014444	001410				BEO	50	;NO
2042	014446	104402	014452			TYPE	..+2	;ASCIZ "DA"
2043	014460	017746	164356			MOV	ORSDA,-(6)	;PUT ORSDA ON STACK
2044	014464	104404				TYPEO		;TYPE STACK IN OCTAL
2045	014466	032737	000010	016040	501	BIS	%NC,.HLTCT	;PRINT NC?
2046	014474	001410				BEO	60	;NO
2047	014476	104402	014502			TYPE	..+2	;ASCIZ "NC"
2048	014510	017746	164322			MOV	ORNC,-(6)	;PUT ORNC ON STACK
2049	014514	104404				TYPEO		;TYPE STACK IN OCTAL
2050	014516	032737	000040	016040	601	BIS	%DS,.HLTCT	;DRIVE STATUS
2051	014524	001410				BEO	70	;NO
2052	014526	104402	014532			TYPE	..+2	;ASCIZ "DS"
2053	014540	017746	164300			MOV	ORSDS,-(6)	;PUT ORSDS ON STACK
2054	014544	104404				TYPEO		;TYPE STACK IN OCTAL
2055	014546	032737	000002	016040	901	BIS	%DB,.HLTCT	;PRINT DATA BUFFER

2056	014554	001461				BEG	PTDONE		;NO
2057	014556	104402	014562			TYPE	..+2		;ASCIZ "DB"
2058	014570	017746	164260			MOV	0RSDB,-(6)		;PUT 0RSDB ON STACK
2059	014574	104404				TYPE0			;TYPE STACK IN OCTAL
2060	014576	000137	014720			JMP	PTDONE		;GET OUT
2061	014602	042737	000200	016040	SEEC:	BIC	0200,.HLTCT		;CLEAR COMMON BIT
2062	014610	032737	000240	016040		BIT	0DT,.HLTCT		;PRINT DRIVE TYPE?
2063	014616	001410				BEG	100		;NO
2064	014620	104402	014624			TYPE	..+2		;ASCIZ "DT"
2065	014632	017746	164222			MOV	0RSDT,-(6)		;PUT 0RSDT ON STACK
2066	014636	104404				TYPE0			;TYPE STACK IN OCTAL
2067	014640	032737	000220	016040	1001	BIT	0MR,.HLTCT		;PRINT MR?
2068	014646	001410				BEG	110		;NO
2069	014650	104402	014654			TYPE	..+2		;ASCIZ "MR"
2070	014662	017746	164170			MOV	0RMR,-(6)		;PUT 0RMR ON STACK
2071	014666	104404				TYPE0			;TYPE STACK IN OCTAL
2072	014670	032737	000204	016040	1101	BIT	0LA,.HLTCT		;PRINT LA?
2073	014676	001410				BEG	PTDONE		;NO
2074	014700	104402	014704			TYPE	..+2		;ASCIZ "LA"
2075	014712	017746	164134			MOV	0RLA,-(6)		;PUT 0RLA ON STACK
2076	014716	104404				TYPE0			;TYPE STACK IN OCTAL
2077	014720	032737	010000	001122	PTDONE:	BIT	0BIT12,FLAG2		;POWER FAIL TEST?
2078	014726	001111				BNE	RETT		;YES
2079	014730	104402	014734			TYPE	..+2		;ASCIZ <15><12>"PASS"
2080	014744	013746	001006			MOV	PCNT+2,-(6)		;PUT PCNT+2 ON STACK
2081	014750	104406				TYPE0			;TYPE STACK IN OCTAL - SUPRESS
2082	014752	032777	001000	164046		BIT	0BIT9,0SHR		;LOOPING ON ERROR?
2083	014760	001404				BEG	DRP		;NO
2084	014762	104402	014766			TYPE	..+2		;ASCIZ <15><12>
2085	014772	032777	000001	164026	DRP:	BIT	0BIT0,0SHR		;DROP DRIVE?
2086	015000	001464				BEG	RETT		;NO
2087	015002	013704	001160			MOV	UNNUM,R4		;GET UNIT 0
2088	015006	004737	014012			JSR	PC,GETERR		;GET ERRORS
2089	015012	063737	001202	001206		ADD	WRITER,READER		;ADD THE ERRORS
2090	015020	063737	001206	001204		ADD	READER,WCERR		
2091	015026	063737	001204	001210		ADD	WCERR,CONERR		
2092	015034	022737	000023	001210		CMP	023,CONERR		;DROP DRIVE?
2093	015042	103043				BNE	RETT		;NO
2094	015044	053737	001164	001124		BIS	UNCMP,DROP		;DROP DRIVE
2095	015052	104402	015056			TYPE	..+2		;ASCIZ <15><12>"DROPPED UNIT"
2096	015076	013746	001160			MOV	UNNUM,-(6)		;PUT UNNUM ON STACK
2097	015102	104406				TYPE0			;TYPE STACK IN OCTAL - SUPRESS
2098	015104	104402	000636			TYPE	,CRLF		
2099	015110	113703	001124			NOVB	DROP,R3		;GET DROPPED UNITS
2100	015114	113704	001162			NOVB	UNITSV,R4		;GET ALL DRIVES
2101	015120	020304				CMP	R3,R4		;ALL DRIVES DROPPED?
2102	015122	001003				BNE	20		;NO
2103	015124	000000				HALT			;NO MORE DRIVES
2104	015126	000137	001234			JMP	00BEGIN		;RESTART TEST
2105	015132	032737	100000	001126	20:	BIT	0BIT15,FLAG		;DID OPERATOR SELECT PATTERN
2106	015140	001002				BNE	30		;YES
2107	015142	005037	001136			CLR	PATNU		;NO CLEAR IT
2108	015146	000137	000666		30:	JMP	00EXTPPR		;GET NEXT DRIVE
2109	015152	000207			RETT:	RTS	PC		

Address	Hex Address 1	Hex Address 2	Hex Address 3	Hex Address 4	Assembly	Comments
2110	015154	013705	015200		ROUTINE TO RESTORE LOADER	
2111					RDR1 MOV LDR1,R5	GET FIRST ADDRESS OF WHERE LOADER IS
2112						TO BE RESTORED
2113	015160	012704	017446		MOV 017446,R4	ADDRESS WHERE LOADER IS STORED
2114	015164	012702	000155		MOV 0155,R2	WORD COUNT
2115	015170	012625			10: MOV (R4)+,(R5)+	RESTORE
2116	015172	005302			DEC R2	
2117	015174	001375			BNE 10	
2118	015176	000000			HALT	DONE
2119	015200	017446			LDR1: .WORD 17446	FIRST ADDRESS WHERE LOADERS ARE SAVED
2120						
2121		172100			PARCSR=172100	
2122		000114			PARVEC=114	
2123	015202	012737	015274	000114	.NANK: MOV 0,PARSRV,00PARVEC	
2124	015210	012737	000340	000116	MOV 0340,00PARVEC+2	SET PRI LEVEL TO 7
2125	015216	013746	000004		MOV 004,-(SP)	SAVE CURRENT ERROR VECTOR
2126	015222	013746	000006		MOV 006,-(SP)	SAVE PRIORITY LEVEL
2127	015226	012737	000006	000004	MOV 06,004	
2128	015234	012737	000002	000006	MOV 0RTI,006	
2129	015242	012700	172100		MOV 0PARCSR,R0	GET FIRST CSR ADDR
2130	015246	012702	000001		MOV 01,R2	
2131	015252	012720	000001		10: MOV 01,(R0)+	SET ACTION ENABLE IF AVAILABLE
2132	015256	006302			ASL R2	SHIFT AVAILABILITY INDICATOR
2133	015260	103374			BCC 10	
2134	015262	012637	000006		MOV (SP)+,006	RESTORE ERROR VECTOR PRIORITY
2135	015266	012637	000004		MOV (SP)+,004	AND INTERRUPT VECTOR
2136	015272	000207			RTS PC	
2137					PARITY MEMORY TRAP	
2138						
2139	015274	104402	000760		.PARSRV: TYPE ,PAREER	
2140	015300	052737	004000	001122	BIS 0BIT11,FLAG2	SET ERROR FLAG
2141	015306	032737	000010	001122	BIS 0BIT3,FLAG2	WERE WE COMPARING DURING ERROR?
2142	015314	001422			BEG 130	NO
2143	015316	104402	015322		TYPE .,+2	ASCII = ADDR=
2144	015332	005737	001212		TST MNAVA	IS MEM MGMT ON?
2145	015336	001406			BEG 120	NO
2146	015340	013746	177776		MOV PS,-(6)	GET PS
2147	015344	013746	001150		MOV SAVE,-(6)	GET VIRTUAL ADDR
2148	015350	104412			TYPEA	CONVERT TO PHY AND TYPE
2149	015352	000403			BR 130	COUNT
2150	015354	013746	001150		120: MOV SAVE,-(6)	GET ADDR
2151	015360	104406			TYPEB	TYPE IT
2152	015362	032777	100000	163436	130: BIT 0BIT15,05WR	HALT ON ERROR?
2153	015370	001401			BEG 10	NO
2154	015372	000000			HALT	YES
2155	015374	012706	000500		10: MOV 0500,SP	RESET STACK
2156	015400	000137	003240		JMP EXNFLG	RESTART TEST

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
2194
2195
2196

015404 010446
015406 010546
015410 017608 000004
015414 032708 177400
015420 001002
015422 016608 000004
015426 105718
015430 001423
015432 122718 000012
015436 001012
015440 113704 001018
015444 113777 001014 163352 501
015452 105777 163340
015456 100375
015460 005304
015462 001370
015464 112577 163334 401
015470 105777 163322
015474 100375
015476 000753
015500 017646 000004
015504 042766 000002 000006
015512 022666 000004
015516 001004
015520 062703 000002
015524 042703 000001
015530 010966 000004
015534 012603
015536 012604
015540 000002

.SDTTL

STYPE - TTY TYPEOUT ROUTINE

;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
;MESSAGE STARTING IN LOCATION "ADR", 2) "TYPE ,CHAR" - TYPES
;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
;IS IN FILCHR+1.

.TYPE:	MOV	R4,-(6)	;SAVE R4
	MOV	R5,-(6)	;SAVE R5
	MOV	04(6),R5	;GET ADDRESS TO BE TYPED
	BIF	0177400,R5	;IS IT A TYPEN?
	BNE	10	;NO
	MOV	4(6),R5	;GET ADDRESS OF CHARACTER
101	TSTB	(R5)	;TERMINATOR?
	BEO	20	;GET OUT IF SO
	CMPS	012,(R5)	;IS THE CHAR A LINE FEED
	BNE	40	;NO - GET OUT
	NOVB	FILCHR+1,R4	;GET THE FILL COUNT
	NOVB	FILCHR,0TPB	;TYPE A FILLER
	TSTB	0TPB	;DONE YET?
	BPL	,=4	;NO - WAIT
	DEC	R4	;DEC COUNT
	BNE	50	;LOOP UNTIL 0
	NOVB	(R5)+,0TPB	;LOAD AND TYPE THE CHARACTER
401	TSTB	0TPB	;IS THE PRINTER READY
	BPL	,=4	;WAIT UNTIL IT IS
	BR	10	;GET THE NEXT CHARACTER
201	MOV	04(6),-(6)	;GET ADDRESS TO BE TYPED
	ADD	02,0(6)	;ADD 2 TO THE ADDRESS
	CMF	(6)+,4(6)	;IS IT .+2?
	BNE	30	;NO
	ADD	02,R5	;ADD 2 TO THE ADDRESS
	BIC	01,R5	;BACK UP TO AN EVEN BYTE
	MOV	R5,4(6)	;RESTORE ADDRESS
301	MOV	(6)+,R5	;RESTORE R5
	MOV	(6)+,R4	;RESTORE R4
	RTI		;RETURN


```

2197          .SBTTL          SSCOPE - SCOPE LOOP HANDLER
2198
2199          ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
2200          ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
2201          ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
2202          ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD;"
2203
2204          015542 104442          .SCOPE: KBDIN          ;GO CHECK FOR "G
2205          015544 032777 040000 163254          BIT      $SW14,$SWR          ;LOOP ON TEST?
2206          015552 001045          BNE      .KIT          ;LOOP ON TEST IS SET
2207          015554 000416          BR       30          ;SKIP - NOP FOR XOR TESTER
2208          015556 013746 000004          MOV      004,-(6)          ;PUSH 004 ON STACK
2209          015562 012737 015602 000004          MOV      040,004          ;SET FOR TIMEOUT
2210          015570 005737 177060          TST      00177060          ;ERROR ON XOR?
2211          015574 012637 000004          MOV      (6)+,004          ;POP STACK INTO 004
2212          015600 000422          BR       .SVLAD          ;NO ERROR - GO TO NEXT TEST
2213          015602 022626          40:      CMP      (6)+,(6)+          ;CLEAR STACK
2214          015604 012637 000004          MOV      (6)+,004          ;POP STACK INTO 004
2215          015610 000426          BR       .KIT          ;ERROR - LOOP ON TEST
2216          015612 032777 004000 163206 30:      BIT      $SW11,$SWR          ;KILL ITERATIONS
2217          015620 001012          BNE      .SVLAD          ;YES - KILL ITERATIONS
2218          015622 105737 001001          TSTB     ICNT+1          ;FIRST ONE?
2219          015626 001404          BEQ      20          ;BRANCH IF FIRST
2220          015630 123737 015714 001001          CNPB     TIMES,ICNT+1          ;DONE?
2221          015636 003013          BGT      .KIT          ;BRANCH IF NOT
2222          015640 112737 000001 001001 20:      MOVB     01,ICNT+1          ;FIRST ITERATION
2223          015646 105237 001000          .SVLAD: INCB     ICNT          ;COUNT TEST NUMBERS
2224          015652 011637 001010          MOV      (6),LAD          ;SAVE LOOP ADDRESS
2225          015656 013777 001000 163144          MOV      ICNT,$DISPLAY          ;DISPLAY TEST NO. AND ITERATION COUNT
2226          015664 000002          RTI          ;RETURN
2227
2228          015666 105237 001001          .KIT:  INCB     ICNT+1          ;INC THE ITERATION COUNT
2229          015672 013777 001000 163130          .OVER: MOV      ICNT,$DISPLAY          ;SET UP DISPLAY
2230          015700 005737 001010          TST      LAD          ;FIRST ONE?
2231          015704 001760          BEQ      .SVLAD          ;YES
2232          015706 013716 001010          MOV      LAD,(6)          ;FUDGE RETURN ADDRESS
2233          015712 000002          RTI          ;FIXES PS
2234
2235          015714 000001          TIMES:  1          ;RUN 1 TIMES

```



```

2236          .SBTTL          SHLT - HLT ROUTINE (ERROR TYPEOUT)
2237
2238          ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
2239          ;ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
2240          ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
2241          ;"HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
2242          ;(HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADDITIONAL TYPEOUTS.
2243
2244          015716 005237 001002          .HLT:      INC      ERRORS          ;INC THE ERROR COUNT
2245          015722 104442          KBDIN          ;GO CHECK FOR "G
2246          015724 032777 020000 163074          BIT      0SW13,0SWR          ;SKIP TYPEOUT IF SET
2247          015732 001025          BNE      20          ;SKIP TYPEOUTS
2248          015734 104402 015740          TYPE      ,,+2          ;,ASCII <15><12>
2249          015744 011637 001012          MOV      (6),HLTADR          ;PUT ADDRESS OF INSTRUCTION ON STACK
2250          015750 162737 000002 001012          SUB      02,HLTADR          ;FUDGE ADDRESS
2251          015756 117737 163030 016040          NOVB     0HLTADR,.HLTCT          ;GET HLT ARGUMENT
2252          015764 013746 001012          MOV      HLTADR,-(6)          ;PUT HLTADR ON STACK
2253          015770 104404          TYPE0          ;TYPE STACK IN OCTAL
2254          015772 104402 015776          TYPE      ,,+2          ;,ASCII " "
2255          016002 004737 014214          JSR      PC,RBREG          ;GO TO USER ERROR ROUTINE
2256          016006 005777 163014          20:      TST      0SWR          ;HALT ON ERROR
2257          016012 100001          BPL      ,+4          ;SKIP IF CONTINUE
2258          016014 000000          HALT          ;HALT ON ERROR!
2259          016016 032777 001000 163002          BIT      0SW9,0SWR          ;CHECK FOR INHIBIT LOOP ON ERROR
2260          016024 001003          BNE      30          ;SKIP IF LOOP ON ERROR
2261          016026 105037 001001          CLRB     ICNT+1          ;CLEAR ITERATION COUNT
2262          016032 000002          RTI          ;RETURN
2263          016034 000137 015666          30:      JNP      .KIT          ;LOOP ON TEST UNTIL NO ERRORS
2264
2265          016040 000000          .HLTCT: 0          ;HLT ARGUMENT

```



```

2266 .SBTTL SOCTAL - OCTAL TYPEOUT ROUTINE
2267
2268 ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
2269 ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
2270 ;16 BITS. IT IS CALLED VIA THE TYOCT, TYPBIT, OR TYPOCS MACRO'S.
2271

2272 016042 012737 170101 016230 .TYPEB: MOV 0170101,.PR ;SET BIT FLAG AND 16. CHARACTER COUNT
2273 016050 000411 .PTIT ;NOW TYPE IT IN BIT FORM
2274 016052 112737 000001 016230 .TYPEO: MOVB 01,.PR ;SET ZERO FILL SWITCH
2275 016060 000402 .BR ;+6 ;SKIP
2276 016062 005037 016230 .TYPES: CLR .PR ;SUPPRESS LEADING ZERO'S
2277 016066 112737 177772 016231 .PTIT: MOVB 0-6,.PR+1 ;SET COUNT
2278 016074
2279 016074 010446 .PTIT: MOV R4,-(6) ;PUSH R4 ON STACK
2280 016076 010546 .PTIT: MOV R5,-(6) ;PUSH R5 ON STACK
2281 016100 016605 000010 .PTIT: MOV 10(6),R5 ;GET THE DATA
2282 016104 012704 016232 .PTIT: MOV 0,PR+2,R4 ;SET POINTER TO FIRST ASCII CHAR.
2283 016110 105014 .PTIT: CLRB (4) ;CLEAR FIRST BYTE
2284 016112 000411 .PTIT: BR .PRF ;ROTATE FIRST BIT
2285 016114 105014 .PRL: CLRB (4) ;CLEAR BYTE OF CHARACTER
2286 016116 032737 000100 016230 .PRL: BIT 0100,.PR ;BIT TYPING MODE?
2287 016124 001004 .PRL: BNE .PRF ;YES - SKIP 2 ROTATES
2288 016126 006105 .PRL: ROL R5 ;ROTATE BIT INTO C
2289 016130 106114 .PRL: ROLB (4) ;PACK IT
2290 016132 006105 .PRL: ROL R5 ;ROTATE BIT INTO C
2291 016134 106114 .PRL: ROLB (4) ;PACK IT
2292 016136 006105 .PRL: ROL R5 ;ROTATE BIT INTO C
2293 016140 106114 .PRL: ROLB (4) ;PACK IT
2294 016142 105714 .PRL: TSTB (4) ;IS IT ZERO?
2295 016144 001402 .PRL: BEQ .+6 ;SKIP INC
2296 016146 105237 016230 .PRL: INCB .PR ;SET FILL SWITCH
2297 016152 105737 016230 .PRL: TSTB .PR ;CHECK FILL SWITCH
2298 016156 001402 .PRL: BEQ .+6 ;SKIP BITSET
2299 016160 152724 000060 .PRL: DISB 0'0,(4)+ ;MAKE INTO ASCII CHAR
2300 016164 105237 016231 .PRL: INCB .PR+1 ;INC COUNT
2301 016170 001351 .PRL: BNE .PRL ;REPEAT
2302 016172 022704 016232 .PRL: CMP 0,PR+2,R4 ;EMPTY BUFFER?
2303 016176 001002 .PRL: BNE .+6 ;SKIP IF NOT
2304 016200 112724 000060 .PRL: MOVB 0'0,(4)+ ;LOAD 1 ZERO
2305 016204 105014 .PRL: CLRB (4) ;NULL TERMINATOR
2306 016206 104402 016232 .PRL: TYPE .,PR+2 ;TYPE IT
2307 016212 012605 .PRL: MOV (6)+,R5 ;POP STACK INTO R5
2308 016214 012604 .PRL: MOV (6)+,R4 ;POP STACK INTO R4
2309 016216 016666 000002 000004 .PRL: MOV 2(6),4(6) ;GET RID OF
2310 016224 012616 .PRL: MOV (6)+,(6) ;DATA WORD
2311 016226 000002 .PRL: RTI ;RETURN
2312
2313 016230 000012 .PRL: .BLKW 12 ;COUNT, SWITCH, AND OUTPUT BUFFER

```



```

2314      .SBTTL      SPower - POWER DOWN AND UP ROUTINES
2315
2316      ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
2317      ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
2318      ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
2319      ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
2320      ;THE PROGRAM WILL HALT AT '.ILLUP'.
2321
2322      016254 012777 016402 000126 .POWER: MOV      0,ILLUP,0,PUEC ;SET FOR FAST UP
2323      016262 012777 000340 000122      MOV      0340,0,PUEC+2 ;PRIO:7
2324      016270 010046      MOV      R0,-(6) ;PUSH R0 ON STACK
2325      016272 010146      MOV      R1,-(6) ;PUSH R1 ON STACK
2326      016274 010246      MOV      R2,-(6) ;PUSH R2 ON STACK
2327      016276 010346      MOV      R3,-(6) ;PUSH R3 ON STACK
2328      016300 010446      MOV      R4,-(6) ;PUSH R4 ON STACK
2329      016302 010546      MOV      R5,-(6) ;PUSH R5 ON STACK
2330      016304 010637 016406      MOV      SP,.SAVR6 ;SAVE SP
2331      016310 012777 016320 000072      MOV      0,PONUP,0,PUEC ;SET UP VECTOR
2332      016316 000000      HALT      ;WAIT FOR PF
2333
2334      016320 013706 016406 .PONUP: MOV      .SAVR6,SP ;GET SP
2335      016324 005001      CLR      R1 ;WAIT LOOP FOR THE TTY
2336      016326 005201 10:      INC      R1 ;WAIT FOR THE INC
2337      016330 001376      BNE      10 ;OF WORD
2338      016332 012605      MOV      (6)+,R5 ;POP STACK INTO R5
2339      016334 012604      MOV      (6)+,R4 ;POP STACK INTO R4
2340      016336 012603      MOV      (6)+,R3 ;POP STACK INTO R3
2341      016340 012602      MOV      (6)+,R2 ;POP STACK INTO R2
2342      016342 012601      MOV      (6)+,R1 ;POP STACK INTO R1
2343      016344 012600      MOV      (6)+,R0 ;POP STACK INTO R0
2344      016346 012737 016254 000024      MOV      0,POWER,0+24 ;SET UP THE POWER DOWN VECTOR
2345      016354 012737 000340 000026      MOV      0340,0+26 ;PRIO:7
2346      016362 104402 016366      TYPE      .,+2 ;.ASCIZ <15><12>"POWER"
2347      016376 000137 007012      JMP      TIMUP ;JMP TO USER ADDRESS
2348
2349      016402 000000 .ILLUP: HALT      ;THE POWER UP SEQUENCE WAS STARTED
2350      016404 000776      BR      .-2 ; BEFORE THE POWER DOWN WAS COMPLETE
2351
2352      016406 000000 .SAVR6: 0 ;PUT THE SP HERE
2353      016410 000024 000026 .PUEC: 24,26 ;POWER UP VECTOR

```


2354
2355
2356
2357
2358
2359
2360
2361 016414
2362 016414 010446
2363 016416 010546
2364 016420 016605 000012
2365 016424 016604 000010
2366 016430 016666 000006 000010
2367 016436 016666 000004 000006
2368 016444 016666 000002 000004
2369 016452 012616
2370 016454 010346
2371 016456 000305
2372 016460 006005
2373 016462 006005
2374 016464 006005
2375 016466 042705 177771
2376 016472 016505 016612
2377 016476 010403
2378 016500 042704 160000
2379 016504 000303
2380 016506 006003
2381 016510 006003
2382 016512 006003
2383 016514 006003
2384 016516 042703 177761
2385 016522 060305
2386 016524 011505
2387 016526 006305
2388 016530 006305
2389 016532 006305
2390 016534 006305
2391 016536 005003
2392 016540 006305
2393 016542 006103
2394 016544 006305
2395 016546 006103
2396 016550 060405
2397 016552 005503
2398 016554 006305
2399 016556 006103
2400 016560 110337 016232
2401 016564 062737 000000 016232
2402 016572 012704 016233
2403 016576 012737 175401 016230
2404 016604 012603
2405 016606 000137 016114
2406
2407 016612 172340
2408 016614 172240
2409 016616 172340

.SDTTL STYPEA - 10 BIT ADDRESS TYPER

;THIS ROUTINE TAKES 2 ARGUMENTS OFF THE STACK (OLD
;SP AND ADDRESS) AND, USING THE MEMORY MANAGEMENT REGISTERS, TYPES
;THE ADDRESS SUPPLIED IN 10 BIT FORM. THIS ROUTINE IS LINKED
;VIA THE 'TYPADR' MACRO.

.TYPEA:

MOV R4,-(6)
MOV R5,-(6)
MOV 12(6),R5
MOV 10(6),R4
MOV 6(6),10(6)
MOV 4(6),6(6)
MOV 2(6),4(6)
MOV (6)+,(6)
MOV R3,-(6)
SWAB R5
ROR R5
ROR R5
ROR R5
BIC 0177771,R5
MOV ,SATAB(5),R5
MOV R4,R3
BIC 0160000,R4
SWAB R3
ROR R3
ROR R3
ROR R3
ROR R3
BIC 0177761,R3
ADD R3,R5
MOV (5),R5
ASL R5
ASL R5
ASL R5
ASL R5
CLR R3
ASL R5
ROL R3
ASL R5
ROL R3
ADD R4,R5
ADC R3
ASL R5
ROL R3
MOVB R3,.PR+2
ADD 0'0,.PR+2
MOV 0,PR+3,R4
MOV 0175401,.PR
MOV (6)+,R3
JMP .PRL

;PUSH R4 ON STACK
;PUSH R5 ON STACK
;R5 - OLD PS WITH PREVIOUS MODE
;R4 - ADDRESS TO BE DECODED AND TYPED
;MOVE
;DOWN
;FOUR
;WORDS
;PUSH R3 ON STACK
;GET THE
;2 PREVIOUS
;MODE BITS
;INTO POSITION
;TO USE AS AN OFFSET
;R5 - SPACE ADDRESS FOR MM
;R3 - REGISTER OFFSET
;CLEAR THE MM REG SELECT BITS
;NOW MAKE
;MM REG
;SELECT BITS
;INTO AN
;OFFSET
;CLEAR THE JUNK BITS
;ADD THE OFFSET TO THE TABLE
;GET THE ISAR DATA
;THIS IS
;TO SHIFT
;THE SEGMENT
;ADDRESS
;INTO AN
;AN 10 BIT
;ADDRESS
;POSITION
;WITH R3 CONTAINING
;THE UPPER 2 BITS
;AND R5 CONTAINING
;THE 10 BIT ADDRESS
;THEN SHIFT FOR TYPING
;GET THE FIRST NUMBER FROM R3
;MAKE IT INTO A NUMBER
;PUDGE IN THE POINTER
;AND THE FLAGS (FILE & S BYTES)
;POP STACK INTO R3
;DECODE AND TYPE THE REST

.SATAB: 172340
172240
172340

;KISAR0
;SISAR0
;KISAR0 - NEVER USED

MAINDEC-11-DERSC-E
DERSCE.P11

RH11-R803-R803/LA-R804 DATA AND RELIABILITY TEST
STYPERA - 10 BIT ADDRESS TYPER

MACY11 27(732) 26-MAY-76 13:00 PAGE 56 SEQ 0071

2410 016620 177640

177640

;UISAR0


```

2411      .BDTTL      STRAP - TRAP HANDLER
2412
2413      ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APPROPRIATE
2414      ;SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
2415      ;THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
2416      ;FOLLOW THIS MACRO.

2417
2418      016622 011646      .TRAP:  MOV      (6),-(6)      ;GET ADDRESS OF TRAP +2
2419      016624 162716 000002      SUB      02,(6)      ;MAKE IT ADDRESS OF TRAP
2420      016630 017616 000000      MOV      0(6),(6)      ;GET TRAP INSTRUCTION
2421      016634 062716 112242      ADD      0,TRP+2-TRAP,(6) ;GET DATA AND MAKE IT AN OFFSET
2422      016640 013607      .TRP:  MOV      0(6)+,PC      ;GO TO PROPER SUBROUTINE
2423
2424      016642 015542      .SCOPE      ;SCOPE      = TRAP+0      (104400)
2425      016644 015404      .TYPE      ;TYPE      = TRAP+2      (104402)
2426      016646 016052      .TYPE0     ;TYPE0     = TRAP+4      (104404)
2427      016650 016062      .TYPES     ;TYPES     = TRAP+6      (104406)
2428      016652 021226      .TYPED     ;TYPED     = TRAP+10     (104410)
2429      016654 016414      .TYPEA     ;TYPEA     = TRAP+12     (104412)
2430      016656 006120      .ERCLR     ;ERCLR     = TRAP+14     (104414)
2431      016660 006142      .DKCHD     ;DKCHD     = TRAP+16     (104416)
2432      016662 017264      .RDOCT     ;RDOCT     = TRAP+20     (104420)
2433      016664 016710      .RDLIN     ;RDLIN     = TRAP+22     (104422)
2434      016666 021172      .UPDAT     ;UPDAT     = TRAP+24     (104424)
2435      016670 000316      .CLRDV     ;CLRDV     = TRAP+26     (104426)
2436      016672 014106      .LOGW      ;LOGW      = TRAP+30     (104430)
2437      016674 014116      .LOGR      ;LOGR      = TRAP+32     (104432)
2438      016676 014124      .LOGWC     ;LOGWC     = TRAP+34     (104434)
2439      016700 014136      .LOGC      ;LOGC      = TRAP+36     (104436)
2440      016702 017174      .CNTLU     ;CNTLU     = TRAP+40     (104440)
2441      016704 017112      .KBDIN     ;KBDIN     = TRAP+42     (104442)
2442      016706 020000      .SUSWR     ;SUSWR     = TRAP+44     (104444)

```



```

2443          .SBTTL          ORDLIN - TTY INPUT ROUTINE
2444
2445          ;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
2446          ;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
2447          ;INSTEAD OF THE RETURN, RUBOUTS ARE HANDLED BY RETYPING
2448          ;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.
2449
2450          .ORDLIN:  MOV     R5, -(6)          ;SAVE R5
2451          10:      MOV     @INPUT, R5        ;GET ADDRESS
2452          20:      CMP     @INPUT+16, R5      ;BUFFER FULL?
2453                   BEQ     40                ;YES - TYPE "?"
2454                   TSTB    @0177560         ;WAIT FOR
2455                   BPL     , -4              ;A CHARACTER
2456                   MOVB    @0177562, (5)     ;GET CHARACTER
2457                   BICB    @200, (5)         ;GET RID OF JUNK
2458                   CNPB    @25, (5)          ;IS IT A "U"
2459                   BNE     50                ;BRANCH IF NOT
2460                   TYPE    ,.+2              ;.ASCIZ "U"<15><12>
2461                   BR      10                ;START OVER
2462          50:      CNPB    @177, (5)          ;IS IT A RUBOUT
2463                   BNE     30                ;SKIP IF NOT
2464          40:
2465                   TYPE    ,.+2              ;.ASCIZ "?"<15><12>
2466                   BR      10                ;ZAP THE BUFFER AND LOOP
2467          30:      MOVB    (5), @0           ;SET UP FOR TYPING
2468                   TYPE    ,.+2              ;ECHO IT
2469                   CNPB    @15, (5)+         ;CHECK FOR RETURN
2470                   BNE     20                ;LOOP IF NOT RETURN
2471                   TYPE    ,12              ;TYPE A LINE FEED
2472                   MOV     (6)+, R5          ;RESTORE R5
2473                   RTI
2474
2475          INPUT:   .BLKB    16.              ;TTY INPUT AREA
2476          ERTAB:   .BLKW    16.
2477
2478          .KBDIN:  TST      42                ;GOT XXDP OR ACT
2479                   BNE OKT                    ;YES, GET OUT
2480                   CMP     @SWREG, SWR        ;GOT SWITCH-LESS MACHINE?
2481                   BNE OKT                    ;NO GET OUT
2482                   TSTB    @TKB              ;HAVE A CHARACTER
2483                   BPL     OKT                ;NO GET OUT
2484                   MOV     @TKB, MSG         ;STRIP OFF GARBAGE
2485                   BIC     @177600, MSG      ;DO WE HAVE A "G"
2486                   CNPB    @7, MSG           ;NO, GET OUT
2487                   BNE OKT                    ;.ASCIZ <15><12>"G"
2488                   TYPE    ,.+2
2489          .CNTLUI: TYPE    ,.+2              ;.ASCIZ <15><12>"SWR"
2490                   MOV     SWREG, -(6)        ;PUT SWREG ON STACK
2491                   TYPEO   ;TYPE STACK IN OCTAL
2492                   TYPE    ,.+2              ;.ASCIZ "  NEW "
2493                   RDOCT
2494                   MOV     (SP)+, MSG        ;GET NEW VALUE OFF STACK
2495                   TST     CTN                ;DID HE TYPE <CR> OF 000000?
2496                   BEQ     OKT                ;DONT CHANGE IF <CR>
2497                   MOV     MSG, SWREG        ;CHANGE VALUE OF SWREG
2498
2499          016710  010546          017032
2500          016712  012705          017052
2501          016716  022705          017052
2502          016722  001423
2503          016724  105737          177560
2504          016730  100375
2505          016732  113715          177562
2506          016736  142715          000200
2507          016742  122715          000025
2508          016746  001006
2509          016750  104402          016754
2510          016762  000753
2511          016764  122715          000177
2512          016770  001005
2513          016772
2514          016772  104402          016776
2515          017002  000743
2516          017004  111527          000000
2517          017010  104402          017006
2518          017014  122725          000015
2519          017020  001336
2520          017022  104402          000012
2521          017026  012605
2522          017030  000002
2523
2524          017032  000020
2525          017052  000020
2526
2527          017112  005737          000042
2528          017116  001057
2529          017120  022737          000176      001026
2530          017126  001053
2531          017130  105777          161664
2532          017134  100050
2533          017136  017737          161660      017260
2534          017144  042737          177600      017260
2535          017152  122737          000007      017260
2536          017160  001036
2537          017162  104402          017166
2538          017174
2539          017174  104402          017200
2540          017210  013746          000176
2541          017214  104404
2542          017216  104402          017222
2543          017234  104420
2544          017236  012637          017260
2545          017242  005737          017262
2546          017246  001403
2547          017250  013737          017260      000176

```


2499 017256 000002
2500
2501 017260 000000
2502 017262 000000
2503
2504

OKT: RTI ;ALL DONE-EXIT
MSG: 0
CTN: 0
SDBTTL SRDOCT - OCTAL INPUT ROUTINE

2505
2506
2507

;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.

2508 017264 011646
2509 017266 016666 000004 000002
2510 017274 010146
2511 017276 010246
2512 017300 010346
2513 017302 104422
2514 017304 005001
2515 017306 005037 017262
2516 017312 012703 017032
2517 017316 112302
2518 017320 122702 000015
2519 017324 001421
2520 017326 122702 000060
2521 017332 003024
2522 017334 122702 000067
2523 017340 002421
2524 017342 006002
2525 017344 006002
2526 017346 006002
2527 017350 006101
2528 017352 006102
2529 017354 006101
2530 017356 006102
2531 017360 006101
2532 017362 005237 017262
2533 017366 000753
2534 017370 010166 000012
2535 017374 012603
2536 017376 012602
2537 017400 012601
2538 017402 000002
2539
2540 017404
2541 017404 104402 017410
2542 017414 000732
2543 017416 000000
2544
2545

.RD OCT: MOV (6),-(6) ;MOVE THE PC
MOV 4(6),2(6) ;MOVE THE PS
MOV R1,-(6) ;PUSH R1 ON STACK
MOV R2,-(6) ;PUSH R2 ON STACK
MOV R3,-(6) ;PUSH R3 ON STACK
40: RD LIN ;READ A LINE INTO INPUT
CLR R1 ;INIT DATA WORD
CLR CTN ;CLEAR COUNT WORD
MOV 0 INPUT,R3 ;INIT POINTER
10: MOVB (3)+,R2 ;GET A BYTE
CMPB 015,R2 ;WAS IT A ,CR?
BEQ 20 ;GET OUT IF YES
CMPB 0'0,R2 ;CHECK FOR 0 OR GREATER
BGT 30 ;ERROR - LESS THAN 0
CMPB 0'7,R2 ;CHECK FOR 7 OR LESS
BLT 30 ;ERROR - GREATER THAN 7
ROR R2 ;GET
ROR R2 ;INTO
ROR R2 ;POSITION
ROL R1 ;FIRST BIT
ROL R2 ;GET
ROL R1 ;SECOND BIT
ROL R2 ;GET
ROL R1 ;THIRD BIT
INC CTN ;YES HE TYPED SOMETHING
BR 10 ;LOOP
20: MOV R1,12(6) ;SAVE THE RESULT
MOV (6)+,R3 ;POP STACK INTO R3
MOV (6)+,R2 ;POP STACK INTO R2
MOV (6)+,R1 ;POP STACK INTO R1
RTI ;RETURN
30: TYPE ,,+2 ;.ASCII '?'<15><12>
BR 40 ;TRY AGAIN
OUTBUF: 0

;NOTE FOR PROGRAMMER***** PROGRAM AT THIS POINT CAN NOT EXCEED A PC OF 17464*****


```

2546          020000          .S 20000
2547          ;NOTE ALL THIS CODE GETS DESTROYED WHEN PATTERN IS WRITTEN
2548
2549 020000 032737 000001 020116 .SUSWR: BIT 0BIT0,SWI
2550 020006 001037 BNE XXX
2551 020010 013746 000006 MOV 6,-(SP) ;SAVE 6 ON STACK
2552 020014 013746 000004 MOV 4,-(SP) ;SAVE 4 ON STACK
2553 020020 012737 020040 000004 MOV 010,4 ;SET UP TRAP ADDRESS
2554 020026 022777 177777 160772 CMP 0-1,0SWR ;TEST 177570
2555 020034 001402 BEQ 20 ;FAKE OUT
2556 020036 000407 BR 30 ;HARDWARE AVAILABLE
2557 020040 022626 10: CMP (SP)+,(SP)+ ;ADJUST STACK
2558 020042 012737 000176 001026 20: MOV 0SWREG,SWR ;SET UP SOFTWARE REGISTERS
2559 020050 012737 000174 001030 MOV 0DISPREG,DISPLAY
2560 020056 022737 000176 001026 30: CMP 0SWREG,SWR ;1ST TIME THRU?
2561 020064 001004 BNE 40 ;NO CHANGE STILL 177570
2562 020066 005737 000042 TST 42 ;ANY XXDP OR ACT
2563 020072 001001 BNE 40 ;SWR000000
2564 020074 104440 CMTLU ;GET INITIAL SETTINGS
2565 020076 012637 000004 40: MOV (SP)+,4 ;REPLACE 4 FROM STACK
2566 020102 012637 000006 MOV (SP)+,6 ;REPLACE 6 FROM STACK
2567 020106 052737 000001 020116 XXX: BIS 0BIT0,SWI ;SET THE BEENHEREBIT
2568 020114 000002 RTI ;ALL DONE
2569
2570 020116 000000 SWI: 0
2571
2572
2573
2574 ;ROUTINE TO SAVE AND LOADER
2575 020120 012700 017776 LDR: MOV 017776,R0
2576 020124 012737 020144 000004 MOV 020,4 ;SET TIME OUT TRAP VECTOR
2577 020132 012737 000340 000006 MOV 0340,6
2578 020140 005720 TST (R0)+
2579 020142 000776 BR -2
2580 020144 022626 20: CMP (SP)+,(SP)+
2581 020146 012737 000006 000004 MOV 06,4
2582 020154 005037 000006 CLR 6
2583 020160 162700 000334 SUB 0334,R0 ;POINT R0 BACK TO LOADER
2584 020164 010037 010200 MOV R0,LDR1 ;SAVE FOR RESTORE ROUTINE
2585 020170 012702 000155 MOV 0155,R2 ;WORD COUNT
2586 020174 012703 017446 MOV 017446,R3 ;WHERE LOADER IS TO BE STORED
2587 020200 012023 10: MOV (R0)+,(R3)+ ;STORE LOADER
2588 020202 005302 DEC R3
2589 020204 001375 BNE 10
2590 020206 000207 RTS PC ;RETURN
2591
2592
2593 ; -A- PORT SIZE
2594
2595 020210 052737 020000 001122 SIZZAP: BIS 0BIT13,FLAG2 ;SET MAPPING BIT
2596 020216 004737 001624 JBR PC,DRVENO ;FIND DRIVE
2597 020222 012737 000002 001224 MOV 02,WORK1 ;START WITH ONE 4K BUFFER
2598 020230 012737 000001 001070 MOV 01,STANEM ;FIRST 4K BANK
2599 020236 012737 007476 001140 MOV 007476,BUF ;GET STARTING ADDR. 5K
2600 020244 012737 000001 001130 MOV 01,WRDCT ;LOAD WC
2601 020252 005037 001134 CLR DRA ;LOAD DA

```


2602	020256	012777	000040	160550		MOV	040,0RSC82	;CLEAR ALL R8 REG	
2603	020264	013777	001160	160542		MOV	UNHUN,0RSC82	;GET DRIVE 0	
2604	020272	012737	000071	001172		MOV	071,CMD	;DO A READ	
2605	020300	104416			40:	DKCMD		;NOW	
2606	020302	105777	160524		10:	TSTB	0RSC81	;DONE YET?	
2607	020306	100375				BPL	10	;NO	
2608	020310	032777	004000	160516		BIT	04000,0RSC82	;DID NEM SET?	
2609	020316	001012				BNE	8121	;YES	
2610	020320	005777	160506			TST	0RSC81	;ANY ERRORS?	
2611	020324	100005				BPL	30	;NO	
2612	020326	012737	000006	001106		MOV	06,SIZEAP	;GET SIZE OF BUFFER	
2613	020334	000137	020526			JMP	0081ZERR	;FOR USER IF HE WISHES IT	
2614	020340	104424			30:	UPDAT		;GET NEXT 4K BANK	
2615	020342	000756				BR	40	;TEST BANK	
2616	020344	005337	001224		8121:	DEC	WORK1	;DEC SIZE OF BUFFER	
2617	020350	013737	001224	001106		MOV	WORK1,SIZEAP	;LOAD SIZE OF A BUFFER	
2618	020356	104402	020362			TYPE	,,+2	;ASCIZ <15><12> "PORT -A- DATA BUFFER	4K TO "
2619	020424	004737	021132			JSR	PC,SIZEPR		


```

2620                                     ; -B- PORT SIZE
2621
2622 020430 012737 000001 001224 SIZEBP: MOV 01,WORK1 ;START WITH ONE 4K BUFFER
2623 020436 012737 037476 001140      MOV 037476,BUF ;GET STARTING ADDR. 4K
2624 020444 012737 000001 001130      MOV 01,WRDCT ;LOAD WC
2625 020452 005037 001134      CLR DWA ;LOAD DA
2626 020456 012777 000040 160350      MOV 040,ORSC82 ;CLEAR ALL RS REG
2627 020464 013777 001160 160342      MOV UNNUM,ORSC82 ;GET DRIVE #
2628 020472 012737 002071 001172      MOV 02071,CMD ;DO A READ
2629 020500 104416      SIZE2: DKCMD ;NOW
2630 020502 105777 160324 101      TSTB 0RSC81 ;DONE YET?
2631 020506 100375      BPL 10 ;NO
2632 020510 032777 004000 160316      BIT 04000,ORSC82 ;DID NEM SET?
2633 020516 001072      BNE SIZE1 ;YES
2634 020520 005777 160306      TST 0RSC81 ;ANY ERRORS?
2635 020524 100047      BPL SIZE3 ;NO
2636 020526      SIZEERR:
2637 020526 104402 020532      TYPE 0,02 ;,ACTS <10><12> WILL NOT CONTINUE TO SIZE MEMORY BECAUS
2638 020616 012737 000006 001110      MOV 06,SIZEBP ;GIVE PROGRAM A BUFFER
2639 020624 104060      HLT 10A10 ;YOU CAN ENTER CONVERSATION MODE
2640 020626 052737 000001 001122      BIS 0BIT0,FLAG2 ;BEEN HERE BEFORE FLAG
2641 020634 042737 020000 001122      BIS 0BIT13,FLAG2 ;CLEAR MAPPING FLAG
2642 020642 000000      HALT ;OR GO TO DERAB
2643 020644 032737 000400 001122 SIZE3: BIT 0BIT0,FLAG2 ;FOUND MEMORY YET?
2644 020652 001011      BNE SIZE3 ;YES
2645 020654 052737 000400 001122      BIS 0BIT0,FLAG2 ;SET FOUND NEM FLAG
2646 020662 013737 001224 001074      MOV WORK1,STDCOM ;GET 1ST BANK
2647 020670 012737 000001 001224      MOV 01,WORK1

```


2648	020676	104424		SIZB3:	UPDAT			;GET NEXT 4K BANK
2649	020700	000677			BR	SIZ2		;TEST BANK
2650	020702	000404			BR	SIZB2		
2651	020704	032737	000400	001122	SIZB1:	BIT0,FLAG2		;FOUND MEMORY?
2652	020712	001771			BEO	SIZB3		;NO
2653	020714	005337	001224		SIZB2:	WORK1		;DEC SIZE OF BUFFER
2654	020720	013737	001224	001110		MOV	WORK1,SIZEBP	;LOAD SIZE OF B BUFFER
2655	020726	032737	000400	001122		BIT	BIT0,FLAG2	;FOUND MEMORY?
2656	020734	001017			BNE	00		;YES
2657	020736	104402	020742		TYPE	,,+2		;ASCIZ <15><12>"NO MEMORY ON -B- PORT"
2658	020772	000447			BR	10		
2659	020774			001				
2660	020774	104402	021000		TYPE	,,+2		;ASCIZ <15><12> "PORT -B- DATA BUFFER"
2661	021030	013737	001074	001224	MOV	STBCON,WORK1		
2662	021036	005001			CLR	R1		
2663	021040	005002			CLR	R2		
2664	021042	004737	021140		JSR	PC,SIEP		
2665	021046	104402	021052		TYPE	,,+2		;ASCIZ " TO "
2666	021060	013737	001110	001224	MOV	SIEBP,WORK1		
2667	021066	063737	001074	001224	ADD	STBCON,WORK1		
2668	021074	005337	001224		DEC	WORK1		
2669	021100	004737	021132		JSR	PC,SIEPR		
2670	021104	052737	000100	001126	DIS	BIT0,FLAG		;SET MULTI PORT FLAG
2671	021112	042737	020000	001122	DIC	BIT13,FLAG2		;CLEAR MAPPING FLAG
2672	021120	052737	000002	001122	DIS	BIT1,FLAG2		;SET BEEN HERE FLAG
2673	021126	000137	001474		JMP	CALH		;CAL BUFFER AND NC
2674								
2675	021132	005001			SIEPR:	CLR	R1	;INIT SETUP
2676	021134	012702	000004		MOV	04,R2		
2677	021140	062701	000001		SIEP:	ADD	01,R1	;SETUP FOR BANK NO
2678	021144	062702	000004		ADD	04,R2		;SETUP FOR SIZE FO MEMORY
2679	021150	020137	001224		CMP	R1,WORK1		;IS THIS THE RIGHT SIZE?
2680	021154	001371			BNE	SIEP		;NO
2681	021156	010246			MOV	R2,-(6)		;PUT R2 ON STACK
2682	021160	104410			TYPED			;TYPE STACK IN DECIMAL
2683	021162	104402	021166		TYPE	,,+2		;ASCIZ "K"
2684	021170	000207			RTS	PC		;RETURN
2685								
2686					;ADD 4K TO TEST ADDR			
2687								
2688	021172	005337	001224		.UPDAT:	INC	WORK1	;INC BANK #
2689	021176	062737	020000	001140	ADD	020000,BUF		;UPDATE BY 4K
2690	021204	022737	177476	001140	CMP	0177476,BUF		;EXCEEDED MEM YET?
2691	021212	001401			BEO	10		;YES
2692	021214	000002			RTI			
2693	021216	062716	000002		101	ADD	02,(6)	;UPDATE RETURN PC BY 2
2694	021222	000002			RTI			
2695								
2696	021224	021226			.TYPED			;TYPED @ TRAP+46 (104446)
2697					.SBTTL	BTYPED = CONVERT		BINARY TO DECIMAL AND TYPE ROUTINE
2698								
2699	021226	012737	100040	021454	.TYPED:	MOV	0100040,.DSIGN	;SET BLANK SWITCH AND SIGN
2700	021234	010046			MOV	R0,-(6)		;PUSH R0 ON STACK
2701	021236	010146			MOV	R1,-(6)		;PUSH R1 ON STACK
2702	021240	010246			MOV	R2,-(6)		;PUSH R2 ON STACK
2703	021242	010346			MOV	R3,-(6)</		

2704	021244	010546			MOV	R5, -(6)	; PUSH R5 ON STACK
2705	021246	012737	100040	021454	MOV	0100040, .DSIGN	; SET BLANK SWITCH AND SIGN
2706	021254	016605	000016		MOV	16(6), R5	; GET DATA TO BE TYPED
2707	021260	100004			BPL	18	; BR IF INPUT IS POS.
2708	021262	005405			NEG	R5	; MAKE THE BINARY NUMBER POS.
2709	021264	112737	000055	021454	MOVB	0'-, .DSIGN	; MAKE THE ASCII NUMBER NEG.
2710	021272	005000		18:	CLR	R0	; ZERO THE CONSTANTS INDEX
2711	021274	012703	021444		MOV	0, DBLK, R3	; SETUP THE OUTPUT POINTER
2712	021300	112723	000040		MOVB	0', (R3)+	; SET THE FIRST CHARACTER TO A BLANK
2713	021304	005002		28:	CLR	R2	; CLEAR THE BCD NUMBER
2714	021306	016001	021434		MOV	.DTBL(R0), R1	; GET THE CONSTANT
2715	021312	160105		38:	SUB	R1, R5	; FORM THIS BCD DIGIT
2716	021314	002402			BLT	40	; BR IF DONE
2717	021316	005202			INC	R2	; INCREASE THE BCD DIGIT BY 1
2718	021320	000774			BR	30	
2719	021322	060105		48:	ADD	R1, R5	; ADD BACK THE CONSTANT
2720	021324	005702			TST	R2	; CHECK IF BCD DIGIT=0
2721	021326	001003			BNE	50	; FALL THROUGH IF 0
2722	021330	105737	021455		TSTB	.DSIGN+1	; STILL DOING LEADING 0'S?
2723	021334	100410			BNI	70	; BR IF YES
2724	021336	106337	021455	58:	ASLB	.DSIGN+1	; MSD?
2725	021342	103003			BCC	60	; BR IF NO
2726	021344	113763	021454	177777	MOVB	.DSIGN, -1(R3)	; YES--SET THE SIGN
2727	021352	052702	000060	68:	BIS	0'0, R2	; MAKE THE BCD DIGIT ASCII
2728	021356	052702	000040	78:	BIS	0', R2	; MAKE IT A SPACE IF NOT ALREADY A DIGIT
2729	021362	110223			MOVB	R2, (R3)+	; PUT THIS CHARACTER IN THE OUTPUT BUFFER
2730	021364	005720			TST	(R0)+	; JUST INCREMENTING
2731	021366	020027	000010		CMF	R0, 010	; CHECK THE TABLE INDEX
2732	021372	002744			BLT	20	; GO DO THE NEXT DIGIT
2733	021374	003002			BGT	00	; GO TO EXIT
2734	021376	010502			MOV	R5, R2	; GET THE LSD
2735	021400	000764			BR	60	; GO CHANGE TO ASCII
2736	021402	105013		88:	CLRB	(R3)	; SET THE TERMINATOR
2737	021404	012605			MOV	(6)+, R5	; POP STACK INTO R5
2738	021406	012603			MOV	(6)+, R3	; POP STACK INTO R3
2739	021410	012602			MOV	(6)+, R2	; POP STACK INTO R2
2740	021412	012601			MOV	(6)+, R1	; POP STACK INTO R1
2741	021414	012600			MOV	(6)+, R0	; POP STACK INTO R0
2742	021416	016666	000002	000004	MOV	2(6), 4(6)	; FUDGE DATA
2743	021424	012616			MOV	(6)+, (6)	; OFF STACK
2744	021426	104402	021444		TYPE	, .DBLK	; NOW TYPE THE NUMBER
2745	021432	000002			RTI		; RETURN
2746							
2747	021434	023420	001750	000144	.DTBL:	10000., 1000., 100., 10.	
2748	021442	000012					
2749	021444	000004			.DBLK:	.BLKW 4	
2750	021454	000000			.DSIGN:	0	
2751							
2752		000001				.END	

ADAM 010676
ADDRR 014144
ADERR 003624
ADHGT 003562
ADDRD 004732
ADRI 004746
ADTL 003316
ADTST 003224
ADT1 003324
ADT1A 003326
AKH 007352
APORT 011350
AS 000100
ASKWC 002544
ASB1 001114
A1 002374
BA 000020
BAD 000000
BEGIN 001234
BELL 000007
BIT0 000001
BIT1 000002
BIT10 002000
BIT11 004000
BIT12 010000
BIT13 020000
BIT14 040000
BIT15 100000
BIT2 000004
BIT3 000010
BIT4 000020
BIT5 000040
BIT6 000100
BIT7 000200
BIT8 000400
BIT9 001000
BLOCK 001154
BLSZ 007556
BPORT 011416
BUF 001140
BUFEXI 007342
BUFEX 007426
BUFSIZ 000674
CALM 001474
CALM1 001500
CHKDV 002150
CHKADT 003650
CHKDAT 012626
CHKDSK 013072
CLEAR 011302
CLIND 014112
CLRDV 104426
CMD 001172

CHDAE 007334
CHPLP 010566
CHPLP1 010632
CHPY 003300
CNTLU 104440
CONDAR 007240
CONERR 001210
CONPAR 010446
CONN 002220
CRLPLP 000636
CTN 017362
DA 000004
DATA 000554
DATAT 003710
DATTES 002454
DB 000002
DISBUF 007050
DISPLA 001030
DISPRE 000174
DKCMD 104416
DKINT 000354
DNA 001134
DONE 014044
DONES 002204
DOWN 012750
DROP 001124
DRP 014772
DRVENO 001624
DS 000040
DSKRD 004570
DT 000240
DYNUN 001664
ELN 004034
ENCLR 104414
ERCOUN 001140
ERRCL 012432
ERRORS 001002
ERTAB 017050
EON 004024
ESH1 004520
EXGEN 010200
EXGEN1 010244
EXNPL 003300
EXRAX 004010
EXRXI 000020
EXTDR 007024
EXTNEN 011472
EXTPN 000114
EXTPPR 000000
EXTT 011414
FILCHR 001014
FILDAT 010014
FLAG 001126

FLAG2 001122
FLAG3 001216
FNDTYP 006540
GETERR 014012
GOOD 000000
HINUM 010200
HINUM1 010350
HISAV 010212
HLT 104000
HLTADR 001012
HRDR 001192
ICNT 001000
INCSER 007214
INPUT 017032
INTEXT 006444
INTFLG 001176
KDDIN 104442
KDPAR0 172360
KDPAR1 172362
KDPAR2 172364
KDPAR3 172366
KDPAR4 172370
KDPAR5 172372
KDPAR6 172374
KDPAR7 172376
KDPAR8 172380
KDPAR10 172382
KDPAR20 172384
KDPAR30 172386
KDPAR40 172388
KDPAR50 172392
KDPAR60 172394
KDPAR70 172396
KIPAR0 172340
KIPAR1 172342
KIPAR2 172344
KIPAR3 172346
KIPAR4 172350
KIPAR5 172352
KIPAR6 172354
KIPAR7 172356
KIPAR8 172360
KIPAR10 172362
KIPAR20 172364
KIPAR30 172366
KIPAR40 172368
KIPAR50 172372
KIPAR60 172374
KIPAR70 172376
LA 000204
LAD 001010
LDAT 004022
LDAT1 004016

LDR 020120
LDR1 015200
LOADSW 000510
LOGC 104436
LOGR 104432
LOGN 104430
LOGWC 104434
LONUM 010204
LONUM1 010346
LOPCNT 001200
LOSAY 010210
MDON 012304
MEN 010352
MHABTO 012306
MHAVA 001212
MHSET 011376
MHVEC 000250
MR 000220
MSTR 004754
MYBYWR 012342
N 000000
NOPORT 002362
NOREC 006520
NONGO 002310
NPRCNT 012426
NPRRET 012410
NPR1 012430
NUNG 001170
OKT 017250
OPDAR 002710
OPDSEL 006500
OPPAT 001020
OPRD 003142
OPNCK 003172
OPNRT 003112
OUT 013446
OUTBUF 017410
PARCEN 172100
PARERR 000760
PARVEC 000114
PASEL 007716
PASEX 010024
PASAC 001154
PATNU 001130
PATO 000254
PAT1 000256
PAT10 000374
PAT11 000274
PAT12 000300
PAT13 000302
PAT14 000304
PAT15 000306
PAT16 000310

PAT17 000312
PAT2 000260
PAT20 000314
PAT3 000262
PAT4 000264
PAT5 000266
PAT6 000270
PAT7 000272
PC 000007
PCNT 001004
PDONN 000743
PFT1 012466
PFT2 013014
PFNAT 012522
PFNATT 012014
PHADDR 001120
PHYCOV 006626
POUFAL 013272
PRIORI 001066
PRNT 014204
PS 177776
PSW 177776
PTDONE 014720
PNDON 012222
PNRPL 012052
PNRPL2 012042
PNRPL3 012042
PNRPL4 012050
PNRPL5 012232
QG 000001
RAND 010240
RANDN 000716
RANDOM 010040
RAND1 010044
RANL 000000
RANEX 000000
RDERR 000007
RDLIN 104422
RDOCT 104420
RDOCT 000020
RDOCT 000044
READER 001204
RECOV 000010
RESTOR 010214
RETT 010102
RLDR 010154
RORL 007640
RORSEC 000010
RREAD 000002
RRR2 011730
RRA3 001000
RRA4 001040
RRC01 001022
RRC02 001024

DERSCE.P11

SYMBOL TABLE

RSDA	001042	SIXERR	020526
RSDB	001054	SIXP	021140
RSDS	001044	SIXPR	021132
RSDT	001060	SIXZAP	020210
RSER	001046	SIXZBP	020430
RSLA	001052	SIX1	020344
RSMR	001056	SIX2	020500
RSREG	014214	SIX3	020644
RSVEC	001062	SLH	004174
RSWC	001036	SLH2	004176
RS04DT	001166	SLH2A	004564
RW	= 000006	SP	= 0000006
RWRED	005706	SR0	= 177572
R0	= 0000000	SR1	= 177574
R1	= 0000001	SR2	= 177576
R2	= 0000002	SR3	= 172516
R3	= 0000003	STADUF	000603
R4	= 0000004	STANEN	001070
R5	= 0000005	STATUS	001064
SANHT	003606	STCON	001074
SAVAST	001072	STHNS	012020
SAVCPU	001076	STTEST	001736
SAVE	001150	SUSWR	= 104444
SAVMGA	001100	SWI	020116
SAVMGB	001102	SWITCH	001174
SAVMGC	001104	SWR	001026
SAVMC	001214	SWRDCI	001144
SAVMCB	001220	SWREG	000176
SCOPE	= 104400	SW0	= 000001
SEADUF	003420	SW1	= 000002
SEEC	014602	SW10	= 002000
SETIND	014130	SW11	= 004000
SETSWI	001230	SW12	= 010000
SHIFT	010072	SW13	= 020000
SHIFT1	010256	SW14	= 040000
SIZB1	020704	SW15	= 100000
SIZB2	020714	SW2	= 000004
SIZB3	020676	SW3	= 000010
SIZEAP	001106	SW4	= 000020
SIZEDP	001110	SW5	= 000040
.	= 021456		

SW6	= 000100
SW7	= 000200
SW8	= 000400
SW9	= 001000
TAG	004630
TAG1	005702
TDNA	001142
TINES	015714
TINUP	007012
TK0	001022
TK8	001020
TPB	001024
TP8	001016
TRACK	001132
TRUERR	006436
TRY	001044
TRYNX	002134
TSTNG	000730
TST1	003324
TST2	003710
TST3	005056
TST4	006064
TYPE	= 104402
TYPEA	= 104412
TYPED	= 104446
TYPEO	= 104404
TYPES	= 104406
TYPREC	011772
UNCNP	001164
UNITST	001162
UNHUN	001160
UNRECO	000522
UNTYT	001746
UP	= 000000
UPCHK	012612
UPDAT	= 104424
UPP	012760
VADDR	001116
VECTRR	006774
WATT	011750

WC	= 000010
WCCON	002600
WCERR	001204
WCKERR	000574
WDCTB	001112
WDERR	010706
WHTHU	006562
WORK	001222
WORK1	001224
WORK2	001226
WRDCMP	010642
WRDCT	001130
WRDINC	010650
WRDNW	013366
WRERR	005270
WRITER	001202
WRLG1	005166
WRKXB	003506
WRACK	000430
WRACK1	005410
WRERR	000564
WRXBL	004160
WRX1	004162
WTNO	011484
XESH	004242
XINCSE	007436
XINCSE	007462
XSEADU	003432
XWAIT	012312
XX	020106
YORN	000641
SENDAD	014070
SEND1	014100
.CLRDV	000316
.CNTLU	017174
.DBLK	021444
.DKCHD	006142
.DSIGN	021484
.DTBL	021434
.ENCLR	006120

.HLT	015716
.HLTCT	010040
.ILLUP	016402
.KBDIN	017112
.KIT	015666
.LOGC	014136
.LOGR	014116
.LOGN	014106
.LOGMC	014124
.MANK	015202
.MSG	017200
.OPDR	000035
.OVER	015672
.PARSR	015274
.POWER	016354
.PONUP	016320
.PR	016230
.PRF	016136
.PRL	016114
.PTIT	016074
.PUVEC	016410
.QG	= 000050
.RDLIN	016710
.RDCT	017344
.SATAS	016612
.SAVRS	016400
.SCOPE	015542
.SUSWR	020000
.SVLAB	015644
.SWOPT	121000
.TBIT	014104
.TRAP	016022
.TRP	016040
.TYPE	015004
.TYPEA	016414
.TYPED	016042
.TYPED	021324
.TYPEO	016052
.TYPES	016002
.UPDAT	021172

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*,RSCE/SOL_DERSCE.P11
 RUN-TIME: 22 22 1 SECONDS
 RUN-TIME RATIO: 07/46=1.0
 CORE USED: 12K (23 PAGES)