

RH11/RS03

DATA RELIABILITY
MD-11-DERSB-B

EP-DERSB-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

Table with multiple columns and rows of data, including headers and numerical values. The content is extremely faint and illegible due to the low contrast of the scan.

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

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97

·
·
i

98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03 AND RS04 DRIVES.

THE DERSB 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 OR RS04) ARE OPERATING CORRECTLY. THIS TEST SHOULD BE USED IN CONJUNCTION WITH THE DERSA DIAGNOSTIC.

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

POP11 STANDARD COMPUTER WITH A MINIMUM OF BK OF MEMORY, AND AN RH11 CONTROLLER WITH AN RS03 OR AN RS04 DISK.

2.2 PRELIMINARY PROGRAMS

DZRSA

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 4
DESCRIPTION

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1 (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 5.1). ALL DOWN FOR WORST CASE

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- PORT FOR ALL MEMORY 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. ALL OF MEMORY WILL BE TESTED. YOU MAY ENTER CONVERSATION MODE AND PUT DATA BUFFER WHERE YOU WISH AND WHAT EVER SIZE YOU WISH.

11.15

175
176
177
178
179
180
181
182
183
184
185
186
167
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

5. OPERATIONAL SWITCH SETTINGS

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<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 TYPEOUT OF RETRY ERRORS
- SW<05> = 1 INHIBIT PASS COUNT
- SW<04> = 1 ALLOWS 8 ERROR TYPEOUTS IN THE COMPARE ROUTINE BEFORE EXECUTING NEXT READ COMMAND. WHEN SWITCH IS 0, ONLY 1 ERROR TYPEOUT IS RECORDED.
- SW<03> = 1 TYPEOUT # OF ERRORS
- SW<02> = 1 INHIBIT MEMORY MANAGEMENT
- SW<01> = 1 DATA TEST ONLY
- SW<00> = 1 DROPS DRIVE AFTER 20 ERRORS

209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 6
DESCRIPTION

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.

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 7
DESCRIPTION

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

244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299

TYPE TO START AT		TYPE TO START AT	
0	000000		
1	020000	20	400000
2	040000	21	420000
3	060000	22	440000
4	100000	23	460000
5	120000	24	500000
6	140000	25	520000
7	160000	26	540000
10	200000	27	560000
11	220000	30	600000
12	240000	31	620000
13	260000	32	640000
14	300000	33	660000
15	320000	34	700000
16	340000	35	720000
17	360000	36	740000
		37	760000

TYPE TO START AT		TYPE TO START AT	
40	1000000	60	1400000
41	1020000	61	14200000
42	1040000	62	1440000
43	1060000	63	1460000
44	1100000	64	1500000
45	1120000	65	1520000
46	1140000	66	1540000
47	1160000	67	1560000
50	1200000	70	1600000
51	1220000	71	1620000
52	1240000	72	1640000
53	1260000	73	1660000
54	1300000	74	1700000
55	1320000	75	1720000
56	1340000	76	1740000
57	1360000	77	1760000

TYPE TO START AT		TYPE TO START AT	
100	2000000	120	2500000
101	2020000	121	25200000
102	2040000	122	2540000
103	2060000	123	2560000

101

300
301

104
105

2200000
2220000

124
125

2600000
2620000

302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345

106	2240000	126	2640000
107	2260000	127	2660000
110	2300000	130	2700000
111	2320000	131	2720000
112	2340000	132	2740000
113	2360000	133	2760000
114	2400000	134	3000000
115	2420000	135	3020000
116	2440000	136	3040000
117	2460000	137	3060000

NOTE: THE FORMULA TO GET NUMBERS
TO BE TYPED IS: EVERY 4K BOUNDARY END
IN FOUR ZEROS
SO DISREGARD THE LAST FOUR DIGITS AND
DIVID THE REMAINING
ADDRESS BY TWO. THE RESULTING NUMBER IS
TO BE TYPED IN FOR
THAT 4K
BANK.

NOTE: TYPE ONLY NUMBERS SHOWN!!!

1. TYPE STARTING 4K BANK # FOR DATA BUFFER ON PORT A
THIS WILL DETERMINE WHERE YOUR 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.

2. HOW MANY 4K BANKS IN DATA BUFFER? (OCTAL)
THIS WILL DETERMINE THE SIZE OF -A- PORT BUFFER. THE SIZE OF
THE DATA BUFFER CAN NOT EXCEED 24K.

K01

MAINDEC-11-DERSB-A
DERSBB.P11 27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST

MACY11 27(1006) 27-OCT-76 11:08 PAGE 11

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 9
DESCRIPTION

EXAMPLE:

```

XXXXXXXXXXXXX 16K
X              X
X  BANK 3  X
X              X
XXXXXXXXXXXXX 12K
X              X
X  BANK 2  X
X              X
XXXXXXXXXXXXX 8K
X              X
X  BANK 1  X
X              X
XXXXXXXXXXXXX 4K
X              X
X  BANK 0  X
X PROGRAM X
XXXXXXXXXXXXX 0

```

```

XXXXXXXX
X      X
X CPU X      BK XXXXX
X      X      XMEMX
XXXXXXXX      0 XXXXX
X              X
X              X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X              X
X              X
X-A-PORT

```

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

TO TEST -A- PORT
 1) 1
 2) 1

346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376

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.

TYPE 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

377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429

DATA PATTERN NO.?

IF NO OPTIONAL DATA PATTERN IS REQUESTED (#22) THE PROGRAM WILL EXECUTE THE FOLLOWING LIST OF DATA PATTERNS.

- 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

430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

NO1

MAINDEC-11-DERSB-A
DERSBB.P11

RH11-RS03 DATA AND RELIABILITY TEST
27-OCT-76 11:03

MACY11 27(1006) 27-OCT-76 11:08 PAGE 14

486

AND COMPARES IT FOR THE CORRECT DATA.

487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 12
DESCRIPTION

RANEX - RANDOM DATA, ADDRESS AND WORD COUNT TEST

THIS ROUTINE TESTS THE ABILITY OF THE SYSTEM TO ACCESS RANDOM ADDRESSES WITH RANDOM DATA. TWO SECTORS OF RANDOM DATA ARE 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.

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.

528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

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

WHERE:

CS1,CS2,ER ETC. = RS11 DISK REGISTERS.
GOOD = EXPECTED DATA.
BAD = DATA RECEIVED.

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

IF SWD 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

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

PASS COMPLETE WILL BE TYPED OUT AT END OF PASS. IT WILL TAKE BETWEEN 15 TO 20 MINUTES TO COMPLETE A PASS. ADD 30 SECONDS FOR EACH 4K. FOR DATA TEST.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 500

MAINDEC-11-DERSB-B RH11-RS03-RS04 DATA RELIABILITY DIAG PAGE 14
DESCRIPTION

8.3 POWER FAIL

THE STARTING ADDRESS FOR THE WRITE POWER FAIL TEST IS 270. A MESSAGE WILL BE TYPED OUT "LOAD SW WITH UNIT # AND CONT." THE OPERATOR NOW HAS TO LOAD THE UNIT # IN OCTAL INTO THE SW REGISTER IN BITS 00-01 AND 02. THEN HIT CONTINUE. 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 274. 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.

%
.TITLE MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST
;COPYRIGHT 1973,1974,1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
;PROGRAM BY STANLEY KARACKIEWICZ

581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001
000000
000046
000052
040000
000200
000137 001234
000210
012706 000500
000214 000137 003226
000220
012706 000500
000224 000137 002346
000230
000137 015164

```

;          SWITCH          USE
;          -----          -----
;          SW15= 100000    ;HALT ON ERROR
;          SW14= 40000     ;LOOP ON FUNCTION
;          SW13= 20000     ;INHIBIT ERROR TYPEOUTS
;          SW12= 10000     ;INHIBIT COMPARISON
;          SW11= 4000      ;HALT ON COMPLETION OF TRANSFER
;          SW10= 2000      ;CONVERSATION MODE
;          SW9= 1000       ;LOOP ON ERROR
;          SW8= 400
;          SW7= 200
;          SW6= 100
;          SW5= 40
;          SW4= 20
;          SW3= 10
;          SW2= 4
;          SW1= 2
;          SW0= 1
;          0
;          46
SENDAD    52
BIT14
;          200
JMP      2#BEGIN    ;START TEST
;          210
MOV      #500,SP    ;SETUP STACK
JMP      2#ADTST    ;RESTART ADDR
;          220
MOV      #500,SP    ;CONVERSATION MODE WITHOUT
JMP      2#A1       ;DATA BUFFER QUESTIONS
;          230
JMP      2#ALDR     ;RESTORE LOADER
```

E02

MAINDEC-11-DERSB-A
DERSBB.P11 27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST

MACY11 27(1006) 27-OCT-76 11:08 PAGE 18

637								
638		000260		.=	260			
639	000260	000137	003302		JMP	2#ADTL		; TRACK AND SECTOR SELECT TEST
640								; WRITE EACH WORD ADDR ON ITSELF AND READ IT BACK
641								; LOCATION 1150 CONTAINS UNIT NO.
642		000264		.=	264			
643	000264	000137	005160		JMP	2#RANEL		; RANDOM ADDRESS, DATA TEST
644								; LOCATION 1150 CONTAINS UNIT NO.
645		000270		.=	270			
646	000270	000137	012356		JMP	2#PFT1		; DISK WRITE POWER FAIL TEST
647		000274		.=	274			
648	000274	000137	012712		JMP	2#PFT2		; DISK WRITE CHECK POWER FAIL TEST

649
650
651 000300 000000
652 000302 177777
653 000304 031463
654 000306 066666
655 000310 100001
656 000312 107070
657 000314 070707
658 000316 052525
659 000320 125252
660 000322 177737
661 000324 146314
662 000326 136363
663 000330 063636
664 000332 000001
665 000334 100005
666 000336 155555
667 000340 133333
668
669
670
671 000342 012777 000040 000464
672 000350 013777 001164 000456
673 000356 000002

:RH11 DATA PATTERNS

PAT0: 0
PAT1: 177777
PAT2: 031463
PAT3: 066666
PAT4: 100001
PAT5: 107070
PAT6: 070707
PAT7: 052525
PAT10: 125252
PAT11: 177737
PAT12: 146314
PAT13: 136363
PAT14: 063636
PAT15: 000001
PAT16: 100005
PAT17: 155555
PAT20: 133333
;PAT21 RANDOM DATA

:CLEAR ALL REGISTERS

.CLR DV: MOV #40, @RSCS2 ;CLEAR ALL REG
MOV UNNUM, @RSCS2 ;GET UNIT #
RTI

.SBTTL

SKMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS

674			
675			
676	177572	SRO=177572	; ADDRESS OF MEM MGMT REGISTER SRO
677	177574	SR1=177574	SR1
678	177576	SR2=177576	SR2
679	172516	SR3=172516	; ADDRESS OF MEM MGMT REGISTER SR3
680			
681	172300	KIPDR0=172300	; ADDRESS OF KERNEL 'I' PAGE
682	172302	KIPDR1=172302	; DESCRIPTOR REGISTERS
683	172304	KIPDR2=172304	
684	172306	KIPDR3=172306	
685	172310	KIPDR4=172310	
686	172312	KIPDR5=172312	
687	172314	KIPDR6=172314	
688	172316	KIPDR7=172316	
689			
690	172320	KDPDR0=172320	; ADDRESSES OF KERNEL 'D' PAGE
691	172322	KDPDR1=172322	; DESCRIPTOR REGISTERS
692	172324	KDPDR2=172324	
693	172326	KDPDR3=172326	
694	172330	KDPDR4=172330	
695	172332	KDPDR5=172332	
696	172334	KDPDR6=172334	
697	172336	KDPDR7=172336	
698			
699	172340	KIPAR0=172340	; ADDRESSES OF KERNEL 'I' PAGE
700	172342	KIPAR1=172342	; ADDRESS REGISTERS
701	172344	KIPAR2=172344	
702	172346	KIPAR3=172346	
703	172350	KIPAR4=172350	
704	172352	KIPAR5=172352	
705	172354	KIPAR6=172354	
706	172356	KIPAR7=172356	
707			
708	172360	KDPAR0=172360	; ADDRESSES OF KERNEL 'D' PAGE
709	172362	KDPAR1=172362	; ADDRESS REGISTERS
710	172364	KDPAR2=172364	
711	172366	KDPAR3=172366	
712	172370	KDPAR4=172370	
713	172372	KDPAR5=172372	
714	172374	KDPAR6=172374	
715	172376	KDPAR7=172376	

H02

MAINDEC-11-DERS9-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 21
DERS99.P11 27-OCT-76 11:03 SKMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS

716				
717	000001	N=	1	; INITIALIZE FOR NEWTST
718	104000	HLT=	EMT	; SET HLT TO EMT FOR ERROR TYPEOUTS
719	177776	PS=	177776	; PROCESSOR STATUS
720	177776	PSW=	PS	; PROCESSOR STATUS WORD
721	177570	SWR=	177570	; SWITCH REGISTER
722	177570	DISPLAY=	SWR	; DISPLAY REGISTER
723	000007	BELL=	7	; BELL
724	000000	R0=	%0	; R0 - DEFINE REGISTERS
725	000001	R1=	%1	; R1
726	000002	R2=	%2	; R2
727	000003	R3=	%3	; R3
728	000004	R4=	%4	; R4
729	000005	R5=	%5	; R5
730	000006	SP=	%6	; R6 - STACK POINTER
731	000007	PC=	%7	; R7 - PROGRAM COUNTER
732	000001	BIT0=	1	; BIT EQUATES
733	000002	BIT1=	2	
734	000004	BIT2=	4	
735	000010	BIT3=	10	
736	000020	BIT4=	20	
737	000040	BIT5=	40	
738	000100	BIT6=	100	
739	000200	BIT7=	200	
740	000400	BIT8=	400	
741	001000	BIT9=	1000	
742	002000	BIT10=	2000	
743	004000	BIT11=	4000	
744	010000	BIT12=	10000	
745	020000	BIT13=	20000	
746	040000	BIT14=	40000	
747	100000	BIT15=	100000	
748	000001	GOOD=	R1	; FOR GOOD DATA
749	000000	BAD=	R0	; FOR BAD DATA

750		000510			. =	510	
751							
752	000510	005015	044524	042515	NOINT:	.ASCIZ	<15><12>"TIMED OUT NO INTERRUPT"
753	000516	020104	052517	020124			
754	000524	047516	044440	052116			
755	000532	051105	052522	052120			
756	000540	000					
757							
758	000541	015	046412	046505	MTRAP:	.ASCIZ	<15><12>"MEM MGMT TRAP"
59	000546	046440	046507	020124			
60	000554	051124	050101	000			
761							
762	000561	015	046012	040517	LOADSW:	.ASCIZ	<15><12>"LOAD SW WITH UNIT # AND CONT"
763	000566	020104	053523	053440			
764	000574	052111	020110	047125			
765	000602	052111	021440	040440			
766	000610	042116	041440	047117			
767	000616	000124					
768							
769	000620	005015	040504	040524	DATA:	.ASCIZ	<15><12>"DATA "
770	000626	000040					
771							
772	000630	051127	052111	020105	WRERR:	.ASCIZ	"WRITE ERR"
773	000636	051105	000122				
774							
775	000642	051127	052111	020105	WCKERR:	.ASCIZ	"WRITE CK ERR"
776	000650	045503	042440	051122			
777	000656	000					
778							
779	000657	122	040505	020104	RDEXR:	.ASCIZ	"READ ERR"
780	000664	051105	000122				
781							
782	000670	005015	040522	042116	RANDOM:	.ASCIZ	<15><12>"RANDOM "
783	000676	046517	000040				
784							
785	000702	005015	042522	047503	RECOV:	.ASCIZ	<15><12>"RECOVERED RETRY CT "
786	000710	042526	042522	020104			
787	000716	042522	051124	020131			
788	000724	052103	000040				
789							
790	000730	005015	047503	046125	NOFND:	.ASCIZ	<15><12>"COULD NOT FIND DRIVE"
791	000736	020104	047516	020124			
792	000744	044506	042116	042040			
793	000752	044522	042526	000			
794	000757	015	005012	000	CRLFLF:	.ASCIZ	<15><12><12>
795							
796	000763	040	054450	047440	YORN:	.ASCIZ	" (Y OR N)"
797	000770	020122	024516	000			
798							
799		000776					.EVEN
800							

```

801          001000          . =      1000
802
803 001000 000000          ICNT:  0          ;LH = ITERATION COUNT ;RH = TEST NO.
804 001002 000000          ERRORS: 0          ;ERROR COUNT
805 001004 000000 000000 PCNT:  0,0        ;2 WORD PASS COUNT
806 001010 000000          LAD:  0          ;LOOP ADDRESS FOR SCOPE
807 001012 000000          HLTADR: 0          ;ADDRESS OF LAST HLT INSTRUCTION EXECUTED
808 001014 001000          FILCHR: 1000        ;FILCHR=0 (CHAR) ;FILCHR+1=2 (COUNT)
809 001016 177564          TPS:  177564       ;OUTPUT STATUS REGISTER
810 001020 177566          TP8:  177566       ;OUTPUT BUFFER
811
812 001022 000000          TEMP1:  0
813 001024 000000          TEMP2:  0
814 001026 000000          TEMP3:  0
815 001030 000000          TEMP4:  0
816
817
818
819
820          ;DISK I/O REGISTERS
821
822 001032 172040          RSCS1: 172040       ;DISK CONTROL + STATUS REGISTER
823 001034 172050          RSCS2: 172050       ;DISK CONTROL + STATUS REGISTER
824 001036 172042          RSWC:  172042       ;WORD COUNT REGISTER
825 001040 172044          RSBA:  172044       ;BUS ADDRESS
826 001042 172046          RSDA:  172046       ;DISK ADDRESS (DESIRED ADDRESS)
827 001044 172052          RSDS:  172052       ;DRIVE STATUS
828 001046 172054          RSER:  172054       ;ERROR REG.
829 001050 172056          RSAS:  172056       ;ATTENTION SUMMARY
830 001052 172060          RSLA:  172060       ;LOOK AHEAD
831 001054 172062          RSD8:  172062       ;DATA BUFFER REGISTER
832 001056 172064          RSMR:  172064       ;MAINTENANCE REGISTER
833 001060 172066          RSDT:  172066       ;DRIVE TYPE REGISTER
834 001062 172070          RSBAE:172070       ;BUS ADDRESS EXTENSION
835 001064 172072          RSCS3: 172072       ;CONTROL AND STATUS 3
836 001066 000204          RSVEC: 204         ;INTERUPT RSVEC
837
838          ;BIT ASSIGNMENTS FOR ERROR TYPE OUTS
839
840          000002          DB=2          ;DATA BUFFER
841          000004          DA=4          ;DESIRED ADD
842          000010          WC=10         ;WORD COUNT
843          000020          BA=20         ;BUS ADDRESS
844          000040          DS=40         ;DRIVE STATUS
845          000100          AS=100        ;ATTENTION SUMMARY
846          000204          LA=204        ;LOOK AHEAD
847          000220          MR=220        ;MAINTENANCE
848          000240          DT=240        ;DRIVE TYPE
849
850 001070 000206          STATUS: 206       ;DISK INTERRUPT STATUS
851 001072 000200          PRIORITY:BIT7     ;DISK PRIORITY LEVEL

```

K02

852	000006	RW=6	;R/W IN PDR REG
853	000000	UP=0	;UP BITY IN PDR REG
854	000250	MMVEC=250	;ADDR OF MEM MGMT ERROR TRAP
855	001074	STAMEM: 0	;STARTING LOC FOR -A- PORT
856	001076	SAVAST: 0	;SAVE LOC FOR STAMEM
857	001100	STBCOM: 0	;STARTING LOC FOR -B- PORT
858	001102	SAVCPU: 0	;SAVE LOC FOR CPUBM
859	001104	SAVMGA: 0	;STARTING ADDR FOR -A- PORT WITH MEM MGMT
860	001106	SAVMGB: 0	;STARTING ADDR FOR B PORT W/MEM MGMT
861	001110	SAVMGC: 0	;STARTING LOC FOR CPU W/MEM MGMT
862	001112	SIZEAP: 0	;SIZE OF A PORT
863	001114	SIZEBP: 0	;SIZE OF B PORT
864	001116	WCOTB: 0	;WC FOR A PORT
865	001120	AOB1: 0	;FLAG FOR PORT BEING TESTED
866	001122	VADDR: 0	;VIRTUAL ADDR
867	001124	PHADDR: 0	;PHYSICAL ADDR
868	001126	FLAG2: 0	;FLAG FOR RESTART AND FOUND DRIVE
869	001130	DROP: 0	;BAD UNITS ON SYSTEM THAT GET DUMPED

;DISCRIPTION OF FLAG2

870		;BIT0 = RESTART
871		;BIT1 = FOUND DRIVE
872		;BIT2 = ERROR DO A CRLF FOR UNIT #
873		;BIT3 = DOING COMPARE
874		;BIT4 = SET A16 IN CS1
875		;BIT5 = SET A17 IN CS1
876		;BIT6 = SET IF MEMORY HAS ALREADY BEEN FOUND
877		;BIT7 = WHEN SET MAKE WC UP TO 28K
878		;BIT8 = FOUND MEMORY ON -B- PORT
879		;BIT9 = POWER DID FAIL
880		;BIT10 = WAITING IN BACKGROUND TEST
881		;BIT11 = PARITY ERROR ROUTINE
882		;BIT12 = POWER FAIL TEST
883		;BIT13 = IN MAP ROUTINE
884		;BIT14 = IN POWER FAIL OR CONVERSATION MODE
885		;BIT15 = ERROR IN POWER FAIL

;DISCRIPTION OF FLAG

886		;BIT0 = USED FOR WRITE COUNTER
887		;BIT1 = USED FOR WRITE COUNTER
888		;BIT2 = TRANSFER MODE 64K
889		;BIT5 = OPTIONAL DMA
890		;BIT6 = TEST -B- PORT
891		;BIT7 = LAST DISK BUFFER FLAG
892		;BIT8 = PROGRAM IS IN ADDRESS OR RANDOM TEST
893		;BIT9 = ERROR DURING TRANSFER
894		;BIT10 = DATA TEST ONLY
895		;BIT11 = MULTIPORT
896		;BIT12 = READ
897		;BIT13 = WRITE CHECK
898		;BIT14 = WRITE
899		;BIT15 = PROGRAM CONTROL MODE
900		
901		
902		
903		
904		
905		

L02

```
906                    ;RH11 DEDICATE REGISTERS (MEMORY)
907
908 001132 000000      FLAG: 0                    ;TEST REGISTER
909 001134 000000      WRDCT: 0               ;WORKING WORD COUNT
910 001136 000000      TRACK: 0              ;WORKING DAE
911 001140 000000      DMA: 0                ;WORKING DAR
912 001142 000000      PATMU: 0             ;DATA PATTERN INDEX
913 001144 000000      BUF: 0               ;WORKING DATA BUFFER (OUT-IN)
914 001146 000000      TDMA: 0              ;TEMP DAR
915 001150 000000      SWRDCT: 0            ;STANDARD WORD COUNT
916 001152 000000      ERCOUNT: 0          ;ERROR COUNT FOR MESSAGES.
917 001154 000000      SAVE: 0             ;
918 001156 000000      HDRER: 0            ;POINTER FOR HARD ERROR
919 001160 000000      BLOCK: 0            ;
920 001162 000000      PASSC: 0            ;
921 001164 000000      UNNUM: 0            ;UNIT CURRENTLY BEING TESTED
922 001166 000000      UNITSV: 0           ;SET BIT=UNIT ON BUS
923 001170 000000      UNCMP: 0            ;FOR COMPARING FOR # OF DEVICE
924 001172 000000      RS04DT: 0          ;FLAG FOR RS04
925 001174 000000      NUMS: 0            ;WORK LOC FOR NUMBER INPUTS
926 001176 000000      CMD: 0             ;LOC FOR CS2 COMMANDS
927 001200 000000      SWITCH: 0          ;FLAG FOR WHICH RANDOM NUMBER GEN
928 001202 000000      INTFLG: 0          ;FLAG FOR INTERRUPT
929 001204 000000      LOPCNT: 0          ;ERROR FLAG AND LOOP COUNTER FLAG
930 001206 000000      WRITER: 0          ;CONTAINS # OF WRITE ERRORS
931 001210 000000      WCERR: 0           ;CONTAINS # OF WRITE CHECK ERRORS
932 001212 000000      READER: 0          ;CONTAINS # OF READ ERRORS
933 001214 000000      COMERR: 0          ;CONTAINS # OF COMPARE ERRORS
934 001216 000000      MMAVA: 0           ;MEM MGMT AVAILABLE INDICATOR
935 001220 000000      SAVWC: 0           ;SAVE LOC FOR CONVERSATION WC ROUTINE
936 001222 000000      FLAG3: 0           ;LOOP IN ADDRESS + RANDOM TST FLAG
937 001224 000000      SAVWCB: 0          ;SAVE WC SIZE FOR -B- PORT
938
939                    ;RH11 WORK REGISTERS
940                    ; (CAN BE CHANGED IN ANY ROUTINE)
941 001226 000000      WORK: 0
942 001230 000000      WORK1: 0
943 001232 000000      WORK2: 0
944                    ERRVEC=4
945                    LERADD=177740
946                    HERADD=177742
947                    MEMERR=177744
```

M02

MAINDEC-11-DERSB-A
DERSBB.P11 27-OCT-76

RH11-RS03 DATA AND RELIABILITY TEST
11:03

MACY11 27(1006) 27-OCT-76 11:08 PAGE 26
SKMMR - KERNAL MEMORY MANAGEMENT REGISTER ASSIGNMENTS

948	001234	012706	000500		BEGIN:	MOV	#500, SP	;SET STACK TO *** 500 ***
949	001240	012737	016372	000024		MOV	#.POWER, @#24	;SET UP PF VECTOR
950	001246	012737	000340	000026		MOV	#340, @#26	;LOCK OUT THE WORLD
951	001254	012737	016036	000030		MOV	#.HLT, @#30	;SET EMT VECTOR
952	001262	012737	000340	000032		MOV	#340, @#32	;LOCK UP
953	001270	012737	016766	000034		MOV	#.TRAP, @#34	;SET TRAP VECTOR
954	001276	012737	000340	000036		MOV	#340, @#36	;LOCK UP
955	001304	005037	001000			CLR	ICNT	;INIT ICNT
956	001310	005037	001010			CLR	LAD	;INIT LAD
957	001314	042737	177677	001132		BIC	#177677, FLAG	;CLEAR FLAG
958	001322	042737	177776	001126		BIC	#177776, FLAG2	;CLEAR ALL EXECPT RESTART
959	001330	005037	001222			CLR	FLAG3	;CLEAR LOOP IN ADDRESS + RANDOM TST FLAG
960	001334	032737	000001	001126		BIT	#BIT0, FLAG2	;IS THIS THE FIRST TIME?
961	001342	001002				BNE	1\$;NO
962	001344	004737	020000			JSR	PC, LDR	;SAVE LOADER
963	001350	000005			1\$:	RESET		;CLEAR THE WORLD
964	001352	012737	000340	177776		MOV	#340, PS	;LOCK UP INTERRUPT LEVELS
965	001360	004537	012322			JSR	R5, ERACL	;CLEAR ERROR COUNTER + PASS CNT
966	001364	005037	001216			CLR	MMAVA	;CLEAR MEM MGMT FLAG
967	001370	005037	001120			CLR	A0B1	;TEST A PORT FIRST
968	001374	032737	000004	177570		BIT	#BIT2, SWR	;WANT MEM MGMT?
969	001402	001021				BNE	3\$;NO
970	001404	012737	001432	000004		MOV	#55, 4	;SET TIMEOUT TRAP
971	001412	012737	000340	000006		MOV	#340, 6	;SET PS
972	001420	005037	177572			CLR	@#SR0	;IS MEM MGMT AVAILABLE?
973	001424	005137	001216			COM	MMAVA	;YES
974	001430	000401				BR	4\$;CONT
975	001432	022626			5\$:	CMP	(6)+, (6)+	;CLEAR STACK
976	001434	012737	000006	000004	4\$:	MOV	#6, 4	;RESET
977	001442	005037	000006			CLR	6	;TRAP
978	001446	032737	000001	001126	3\$:	BIT	#BIT0, FLAG2	;IS THIS THE FIRST TIME
979	001454	001002				BNE	CALM	;NO
980	001456	000137	020070			JMP	SIZZAP	;SIZE BUFFERS
981	001462	004737	011514		CALM:	JSR	PC, @#EXTMEM	;SET UP DATA BUFFERS
982	001466	004737	015212		CALM1:	JSR	PC, MAMK	;TURN ON PARITY MEM
983	001472	032737	000001	001126		BIT	#BIT0, FLAG2	;1ST TIME ?
984	001500	001003				BNE	3\$;NO
985	001502	013737	001150	001220		MOV	SWRDC, SAVWC	;SAVE WC FOR CONVERSATION MODE COMPARE
986	001510	052737	000001	001126	3\$:	BIS	#BIT0, FLAG2	;SET 1ST TIME FLAG
987	001516	005037	001140			CLR	DMA	;CLEAR DAR REGISTERS
988	001522	005037	001142			CLR	PATNU	;CLEAR PATTEN COUNT
989	001526	013737	001150	001134		MOV	SWRDC, WRDC	
990	001534	032737	000002	177570		BIT	#BIT1, SWR	;DATA TEST ONLY?
991	001542	001403				BEQ	2\$;NO
992	001544	052737	002000	001132		BIS	#BIT10, FLAG	;YES
993	001552	032737	002000	177570	2\$:	RIT	#BIT10, SWR	;ENTER CONVERSATION MODE?
994	001560	001015				BNE	1\$;YES GO TO CONVERSATION MODE
995	001562	052737	074000	001132		BIS	#74000, FLAG	
996	001570	004537	010244			JSR	R5, RESTOR	;RESTORE ORIGINAL WD CNT
997	001574	012737	017450	000004		MOV	#TIMEOUT, ERVVEC	
998	001602	012737	000340	000006		MOV	#340, ERVVEC+2	
999	001610	000137	003226			JMP	ADTST	
1000	001614	000137	002220		1\$:	JMP	@#CONM	;ENTER CONVERSATION MODE

```

1001 ;FIND OUT HOW MANY DRIVES
1002 ;FIRST TEST RSAS
1003
1004 001620 012701 000010 DRVENO: MOV #8, R1 ;PUT 8 INTO R1 FOR COUNT
1005 001624 042737 000002 001126 BIC #BIT1, FLAG2 ;CLEAR FOUND DRIVE FLG
1006 001632 012777 000000 177174 MOV #0, @RSCS2 ;SET DEVICE TO ZERO
1007 001640 012777 000007 177200 TRY: MOV #7, @RSER ;CAUSE AN ERROR +SETS BIT IN AS REG
1008 001646 005301 DEC R1 ;DO A MAXIMUM OF 16 TIMES
1009 001650 001403 BEQ DVNUM ;TESTED FOR ALL DRIVES GET OUT
1010 001652 005277 177156 INC @RSCS2 ;INCREMENT DRIVE UNIT
1011 001656 000770 BR TRY ;REPEAT FOR NEXT DRIVE
1012 001660 017737 177164 001166 DVNUM: MOV @RSAS, UNITSV ;SAVE
1013 001666 043737 001130 001166 BIC DROP, UNITSV ;DROP BAD DRIVES
1014 001674 012737 000401 001170 MOV #401, UNCOMP ;SETUP TO CMP WITH UNITSV
1015 001702 012737 000000 001164 MOV #0, UNNUM ;PUT 0 INTO UNIT NO.
1016 001710 032737 000040 177570 BIT #BITS, SWR ;INHIBIT TYPE OUT?
1017 001716 001015 BNE STTEST ;YES
1018 001720 104402 001724 TYPE ;ASCIZ <15><12>"TESTING UNIT "
1019 001744 042737 000004 001126 BIC #BIT2, FLAG2 ;CLEAR ERROR FLAG
1020 001752 033737 001170 001166 STTEST: BIT UNCOMP, UNITSV ;IS THIS DRIVE ON THE SYSTEM
1021 001760 001463 BEQ TRYNX ;NO
1022 001762 013777 001164 177044 UNTYP: MOV UNNUM, @RSCS2 ;YES PUT UNIT # INTO CS2
1023 001770 005037 001172 CLR RSO4DT ;CLEAR DRIVE TYPE FLAG
1024 001774 005777 177060 TST @RSDT ;IS THIS A RS03?
1025 002000 001417 BEQ 1$ ;YES
1026 002002 022777 000001 177050 2$: CMP #1, @RSDT ;IS THIS A RS03 4US?
1027 002010 001413 BEQ 1$ ;YES
1028 002012 022777 000002 177040 3$: CMP #2, @RSDT ;IS THIS A RS04?
1029 002020 001404 BEQ 6$ ;YES
1030 002022 022777 000003 177030 CMP #3, @RSDT ;RS04?
1031 002030 001037 BNE TRYNX ;GET A NEW NUMBER
1032 002032 052737 177777 001172 6$: BIS #-1, RSO4DT ;YES RS04
1033 002040 032737 040000 001126 1$: BIT #BIT14, FLAG2 ;IN POWER FAIL OR CONVERSATION?
1034 002046 001401 BEQ 7$ ;NO
1035 002050 000207 RTS PC ;YES
1036 002052 032777 000200 176764 7$: BIT #BIT7, @RSDS ;IS THIS DRIVE READY ?
1037 002060 001423 BEQ TRYNX ;NO GET ANOTHER DRIVE
1038 002062 032737 000040 177570 BIT #BITS, SWR ;TYPEOUT?
1039 002070 001016 BNE 4$ ;NO
1040 002072 032737 000004 001126 BIT #BIT2, FLAG2 ;WAS THERE AN ERRER?
1041 002100 001402 BEQ 5$ ;NO
1042 002102 104402 000757 TYPE , CRLF
1043 002106 5$:
1044 002106 013746 001164 MOV UNNUM, -(6) ;PUT UNNUM ON STACK
1045 002112 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
1046 002114 104402 000040 TYPE , 40 ;TYPE SPACE
1047 002120 042737 000004 001126 BIC #BIT2, FLAG2 ;CLEAR ERROR FLAG
1048 002126 000430 4$: BR NOWGO ;NOW TEST
    
```

B03

```

1049 002130 006337 001170      TRYNX:  ASL      UNCMP      ;CHECK NEXT BIT FOR DRIVE
1050 002134 103403              BCS      CHCKDV     ;DID WE TEST ANY REG?
1051 002136 005237 001164      INC      UNNUM      ;INC UNIT #
1052 002142 000703              BR       STTEST     ;CHECK FOR NEXT DRIVE
1053
1054 002144 032737 000002 001126  CHCKDV: BIT      #BIT1,FLAG2   ;FOUND DRIVE?
1055 002152 001014              BNE     DONEE      ;YES WE DID TEST A DRIVE
1056 002154 012737 100000 001170  MOV     #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
1057 002162 005037 001164      CLR     UNNUM      ;ANY AS BITS, THUS DEFAULTS TO 0
1058 002166 013746 001164      MOV     UNNUM,-(6)  ;PUT UNNUM ON STACK
1059 002172 104406              TYPES   ;TYPE STACK IN OCTAL - SUPRESS
1060 002174 104402 000730      TYPE   ,NOFND
1061 002200 000000              HALT
1062
1063
1064 002202 000402              BR       NOWGO
1065 002204 000137 013366      DONEE:  JMP      OUT
1066 002210 052737 000002 001126  NOWGO:  BIS      #BIT1,FLAG2 ;FOUND DRIVE
1067 002216 000207              RTS      PC
1068
1069      ;ENTER OPERATOR CONVERSATION MODE
1070
1071 002220 104402 006616      CONM:   TYPE     ,STABUF
1072 002224 104420              RDOCT
1073 002226 012637 001074      MOV     (6)+,STAMEM ;START BUFFER AT 4K
1074 002232 104402 006643      3$:    TYPE     ,BUFSIZ
1075 002236 104420              RDOCT
1076 002240 011637 001112      MOV     (6),SIZEAP  ;GET NUMBER
1077
1078      ;XYZ*****?*****
1079      ;*****
1080      ;XYZ*****?*****
1081      ;*****
1082      ;XYZ*****?*****
1083      ;*****
1084      ;XYZ*****?*****
1085      ;*****
1086 002244 063716 001074      ADD     STAMEM,(6)  ;GET STARTING BANK
1087 002250 023726 001022      CMP     TEMPI,(6)+ ;IS TOTAL MEMORY MORE THAN LAST BANK
1088 002254 002766              BLT     3$         ;GREATER THEN 24K
1089 002256 000137 002334      JMP     NOPORT     ;GET OUT

```

C03

Line	Address	Target	Value	Label	Operation	Comments	
1090	002262	104402	006616	1S:	TYPE	,STABUF	
1091	002256	104402	002272		TYPE	,.+2	
1092	002276	104420			RDOCT	;ASCIZ "B "	
1093	002300	012637	001100		MOV	(6)+,STBCOM	
1094	002304	104402	006643	4S:	TYPE	,BUFSIZ	
1095	002310	104420			RDOCT	;GET ANS	
1096	002312	012637	001114		MOV	(6)+,SIZEBP	
1097	002316	022737	000006	001114	CMP	#6,SIZEBP	
1098	002324	002767			BLT	4S	
1099	002326	052737	000100	001132	BIS	#BIT6,FLAG	
1100	002334	004737	011514	NOPORT:	JSR	PC,EXTMEM	
1101	002340	013737	001150	001134	MOV	SWDOCT,WRDOCT	
1102	002346	052737	002000	001132	AI:	BIS	#BIT10,FLAG
1103	002354	004537	012322		JSR	RS,ERRCL	
1104	002360	042737	174040	001132	BIC	#174040,FLAG	
1105	002366	104402	002372		TYPE	,.+2	
1106	002410	004737	003264		JSR	PC,CMFY	
1107	002414	001004			BNE	DATTES	
1108	002416	052737	004000	001132	BIS	#BIT11,FLAG	
1109	002424	000444		1S:	BR	ASKWC	
1110	002426			DATTES:			
1111	002426	104402	002432		TYPE	,.+2	
1112	002452	104420			RDOCT	;ASCIZ <15><12>"TYPE UNIT # "	
1113	002454	012637	001174		MOV	(6)+,NUMS	
1114	002460	022737	000010	001174	CMP	#10,NUMS	
1115	002466	103757			BLO	DATTES	
1116	002470	013737	001174	001164	MOV	NUMS,UNNUM	
1117	002476	004737	006770		JSR	PC,FNDTYP	
1118	002502	005002		1S:	CLR	R2	
1119	002504	000261			SEC		
1120	002506	006102		2S:	ROL	R2	
1121	002510	005737	001174		TST	NUMS	
1122	002514	001403			BEG	3S	
1123	002516	005337	001174		DEC	NUMS	
1124	002522	000771			BR	2S	
1125	002524	010237	001166	3S:	MOV	R2,UNITSV	
1126	002530	052737	000002	001126	BIS	#BIT1,FLAG2	
1127							
1128	002536			ASKWC:			
1129	002536	104402	002542		TYPE	,.+2	
1130	002564	004737	003264		JSR	PC,CMFY	
1131	002570	001403			BEG	WCCON	
1132	002572	004537	010244		JSR	RS,RESTOR	
1133	002576	000431			BR	OPDAR	

D03

Address	OpCode	Source	Target	Comment
1134	002600			WCCON: TYPE ,.+2 ;.ASCIZ <15><12>"WD CT "
1135	002600	104402	002604	RDOCT
1136	002616	104420		
1137	002620	012637	001174	MOV (6)+,NUMS ;GET NUMBER
1138	002624	005737	001174	TST NUMS ;IS IT 0?
1139	002630	001763		BEG WCCON ;YES ASK AGAIN FOR LENGTH
1140	002632	013702	001220	MOV SAVWC,R2 ;GET STANDARD WC FOR -A- PORT
1141	002636	005202		IS: INC R2
1142	002640	020237	001174	CMP R2,NUMS ;IS NUMS LESS THAN SWRDCT
1143	002644	101755		BLOS WCCON ;YES ASK FOR COUNT AGAIN
1144	002646	013737	001174 001150	MOV NUMS,SWRDCT ;OPERATING WORD COUNT
1145	002654	013737	001150 001134	MOV SWRDCT,WRDCT
1146				
1147	002662			OPDAR: TYPE ,.+2 ;.ASCIZ <15><12>"OPTIONAL DSK ADDR"
1148	002662	104402	002666	JSR PC,CMPY ;COMPARE FOR YES
1149	002712	004737	003264	BNE OPPAT ;ANS IS NO
1150	002716	001034		BIS #BITS,FLAG ;SET OPTIONAL DMA FLAG
1151	002720	052737	000040 001132	IS: TYPE ,.+2 ;.ASCIZ <15><12>"DSK ADDR "
1152	002726	104402	002732	RDOCT
1153	002746	104420		
1154	002750	012637	001174	MOV (6)+,NUMS ;GET NUMBER
1155	002754	005737	001172	TST RS04DT ;IS THIS A RS04?
1156	002760	001404		BEG IS ;NO
1157	002762	022737	017777 001174	CMP #17777,NUMS ;IS ADD. CORRECT?
1158	002770	000403		BR 2S ;GET OUT
1159	002772	022737	007777 001174	IS: CMP #7777,NUMS ;IS ADD. CORRECT?
1160	003000	101730		2S: BLOS OPDAR ;NO
1161	003002	013737	001174 001146	MOV NUMS,TDMA ;TEMP SECTOR REGISTER

E03

Line	Address	OpCode	Target	Label	OpCode	Comments
1162	003010				OPPAT:	
1163	003010	104402	003014		TYPE	;.ASCIZ <15><12>"DATA PATTERN # "
1164	003036	104420			RDOCT	
1165	003040	012637	001174		MOV	(6)+,NUMS ;GET NUMBER
1166	003044	022737	000023	001174	CMP	#23,NUMS ;TEST FOR CORRECT NO
1167	003052	101756			BLOS	OPPAT ;ASK AGAIN
1168	003054	005037	001142		CLR	PATNU ;CLEAR PATTERN #
1169	003060	022737	000022	001174	CMP	#22,NUMS
1170	003066	001411			BEQ	OPWRT ;DATA PATTERN UNDER PROGRAM CONTROL
1171	003070	052737	100000	001132	BIS	#BIT15,FLAG ;SET PROGRAM FLAG
1172	003076	013737	001174	001142	MOV	NUMS,PATNU ;OPERATOR WANTS TO SELECT DATA
1173	003104	000241			CLC	
1174	003106	006137	001142		ROL	PATNU
1175						
1176	003112				OPWRT:	
1177	003112	104402	003116		TYPE	;.ASCIZ <15><12>"WRITE"
1178	003126	004737	003264		JSR	PC,CMFY ;COMPARE FOR YES
1179	003132	001003			BNE	OPRD ;ASK ABOUT WRITE CHECK
1180	003134	052737	040000	001132	BIS	#BIT14,FLAG ;YES SET FLAG BIT
1181						
1182	003142				OPRD:	
1183	003142	104402	003146		TYPE	;.ASCIZ <15><12>"READ"
1184	003156	004737	003264		JSR	PC,CMFY ;COMPARE FOR YES
1185	003162	001003			BNE	OPWCK
1186	003164	052737	010000	001132	BIS	#BIT12,FLAG ;SET FLAG TO READ
1187						
1188	003172				OPWCK:	
1189	003172	104402	003176		TYPE	;.ASCIZ <15><12>"WRITE CK"
1190	003212	004737	003264		JSR	PC,CMFY ;COMPARE FOR YES
1191	003216	001003			BNE	ADTST
1192	003220	052737	020000	001132	BIS	#BIT13,FLAG
1193						
1194	003226	032737	004000	001132	ADTST:	BIT ;ARE WE IN MULTI DRIVE MODE
1195	003234	001402			BEQ	EXMFLG ;BRANCH IF NO.
1196	003236	004737	001620		JSR	PC,DRVENO ;GET DRIVES TO BE TESTED
1197	003242	042737	000004	001132	EXMFLG:	BIC ;CLEAR XFER MODE FLAG
1198	003250	032737	002000	001132	IS:	BIT ;TEST FOR DATA TEST ONLY
1199	003256	001414			BEQ	ADT1 ;DO COMPLETE TEST
1200	003260	000137	003710		JMP	DATAT ;DO DATA TEST ONLY
1201						
1202	003264	104402	000763		CMFY:	TYPE ;YORN
1203	003270	104422			RDLIN	
1204	003272	122737	000131	017270	CMFB	#'Y,INPUT ;TEST FOR YES
1205	003300	000207			RTS	PC
1206	003302	052737	100000	001222	ADTL:	BIS ;SET LOOP IN ADDRESS TEST FLAG GOT HERE
1207						;BECAUSE PROGRAM WAS STARTED AT 260

F03

```

1208
1209
1210
1211
1212
1213
1214
1215
1216 003310
1217
1218
1219
1220 003310 104400
1221 003312 012737 000340 177776
1222 003320 005037 001024
1223 003324 012737 020000 017350
1224 003332 052737 000400 001132
1225 003340 013737 001150 001154
1226 003346 005037 001140
1227 003352 104426
1228 003354 012737 000200 001134
1229 003362 012737 000200 001150
1230 003370 005737 001172
1231 003374 001006
1232 003376 012737 000100 001134 15:
1233 003404 012737 000100 001150
1234 003412 013737 017350 001144 25:
1235 003420 104414 SEABUF:
1236 003422 013700 017350
1237 003426 013701 001134
1238 003432 013720 001140 XSEABUF:
1239 003436 005301
1240 003440 001374
1241 003442 012737 000061 001176
1242 003450 104416
1243 003452 105777 175354
1244 003456 100375
1245 003460 005777 175346
1246 003464 100010
1247 003466 012737 003420 001010
1248 003474 052737 001000 001132
1249 003502 104430
1250 003504 104034
1251 003506 104400
1252 003510 004737 007234
1253 003514 000741
1254 003516 104400
  
```

```

;RH11 ADDRESS TEST #1 (TRACK AND SECTOR SELECTION TEST)
;WRITE 100(OCTAL) RS03, 200(OCTAL) RS04, WORDS IN EACH SECTOR
;THE WORD CONTAINS THE ADDRESS OF EACH SECTOR
;WHEN THE COMPLETE DISK IS WRITTEN READ
;BACK EACH SECTOR AND COMPARE FOR THE CORRECT
;DATA IN THE SECTOR
;PS IS AT LEVEL 7 SO NO INTERRUPTS

ADT1: ;ADDRESS TEST
;*****
;TEST 1 ADDRESS TEST
;*****
TST1: SCOPE
ADT1A: MOV #340,PS ;LOCK UP PS
CLR TEMP2
MOV #20000,OUTBUF ;START BUF AT 20000
BIS #BIT8,FLAG ;SET TEST FLAG
MOV SWRDCT,SAVE ;SAVE STD WD COUNT
CLR DMA ;CLEAR DISK ADD
CLR DV ;INIT DRIVE
MOV #200,WRDCT ;SETUP WC
MOV #200,SWRDCT
TST RS04DT ;IS THIS A RS04?
BNE 25 ;YES
15: MOV #100,WRDCT ;SETUP WORD COUNT
MOV #100,SWRDCT
25: MOV OUTBUF,BUF ;SET UP CURRENT ADDRESS
SEABUF: ERCLR ;CLEAR RS REGISTERS IF ERROR
MOV OUTBUF,R0 ;SET UP ADDRESS BUFFER
MOV WRDCT,R1
XSEABUF: MOV DMA,(0)+ ;LOAD OUTBUF WITH DATA TO BE WRITTEN
DEC R1 ;FILL OUTBUF
BNE XSEABUF ;WITH DATA
MOV #61,CMD ;WRITE NO I/E
DKCMD ;GO WRITE
TSTB @RSCS1 ;CHECK FOR READY
BPL -4
TST @RSCS1 ;TEST FOR ERROR
BPL WRNEXB ;BRANCH IF NO ERROR
MOV #SEABUF,LAD ;SET UP LOOP ADDRESS
BIS #BIT9,FLAG ;SET ERROR BIT IN FLAG
LOGW ;LOG WRITE ERROR
HLT !WC!DA!BA
WRNEXB: SCOPE
JSR PC,DISBUF ;SET UP NEXT DISK ADDR.
BR SEABUF ;WRITE NEXT SECTOR
RNDSEC: SCOPE
  
```



```

1255 003520 104414          ROSECT: ERCLR          ;CLEAR ERRORS
1256 003522 012737 000071 001176      MOV      #71,CMD      ;READ NO I/E
1257 003530 104416          DKCMD              ;DO A READ
1258 003532 105777 175274          TSTB     @RSCS1      ;CHECK FOR READY
1259 003536 100375          BPL      -4          ;NOT READY BRANCH BACK
1260 003540 005777 175266          TST      @RSCS1      ;TEST FOR ERROR
1261 003544 100006          BPL      ADHGT       ;BRANCH IF NO ERROR
1262 003546 052737 001000 001132      BIS      #BIT9,FLAG  ;SET ERROR FLAG
1263 003554 104432          LOGR              ;LOG READ ERROR
1264 003556 104014          HLT      !MC!DA
1265 003560 104400          SCOPE
1266 003562 013702 017350          ADHGT: MOV      OUTBUF,R2
1267 003566 005737 001172          TST      RS04DT      ;RS04?
1268 003572 001403          BEQ      1$         ;NO
1269 003574 012703 000200          MOV      #200,R3     ;YES
1270 003600 000402          BR       SANHT       ;CONT
1271 003602 012703 000100          1$:  MOV      #100,R3
1272 003606 023712 001140          SANHT: CMP      DMA,(2) ;CMP FOR CORRECT ADDR.
1273 003612 001004          BNE      ADERR      ;BRANCH IF DATA DID NOT COMPARE
1274 003614 005722          TST      (2)+       ;GET NEXT ADDRESS OF INBUF
1275 003616 005303          DEC      R3         ;DEC SECTOR COUNT
1276 003620 001372          BNE      SANHT       ;TEST NEXT WORD
1277 003622 000412          BR
1278 003624 013701 001140          ADERR: MOV      DMA,GOOD ;CORRECT ADDRESS
1279 003630 011200          MOV      (2),BAD    ;DATA IN ERROR
1280 003632 104000          HLT              ;DISK ADD DID NOT MATCH WRITTEN ADDRESS
1281 003634 104436          LOGC              ;LOG COMPARE ERROR
1282 003636 004737 014134          JSR      PC,PRNT    ;INHIBIT TYPEOUT?
1283 003642 001002          BNE      CHKADT     ;YES
1284 003644 104402 000757          TYPE      ,CRLF
1285
1286          ;*****REPORT ONLY ONE ERROR PER SECTOR*****
1287
1288 003650 104400          CHKADT: SCOPE
1289 003652 004737 007234          JSR      PC,DISBUF  ;SET UP NEXT DISK BUFFER
1290 003656 000717          BR       RR0SEC     ;CHECK NEXT SECTOR
1291 003660 013737 001154 001150          MOV      SAVE,SARDCT ;GET STD WD COUNT
1292 003666 042737 000400 001132      BIC      #BIT8,FLAG ;CLEAR TEST FLAG
1293 003674 032737 100000 001222      BIT      #BIT15,FLAG3 ;DOES OPERATOR WANT TO LOOP ON TEST
1294 003702 001402          BEQ      +6         ;NO
1295 003704 000137 003312          JMP      ADT1A      ;YES
1296 003710
1297          DATAT:          ;DATA TEST
1298          ;*****
1299          ;TEST 2          DATA TEST
1300          ;*****
1301          TST2:  SCOPE
1302          ;XYZ*****?*****
1303          ;*****
1304          ;XYZ*****?*****
1305          ;*****
1306          ;XYZ*****?*****
1307          ;*****
1308          ;XYZ*****?*****
1309          ;*****
1310          MOV      #-1,TEMP2 ;TEST 2 INDICATOR
1311          MOV      STAMEN,TEMP3

```

H03

1311	003726	013737	001150	001134	LDAT3:	MOV	SWR0CT,WRDCT	
1312	003734	005037	001140		LDAT2:	CLR	DMA	;CLEAR DISK ADDRESS
1313	003740	104426				CLRDV		;CLEAR RS REGISTERS
1314	003742	004737	011426			JSR	PC,APORT	
1315	003746	004737	007160		2\$:	JSR	PC,VECTRR	;SETUP INT VECTOR
1316	003752	012777	000340	175110		MOV	#340,STATUS	;SET DISK STATUS REG LOC (206)
1317	003760	012737	004012	001156		MOV	#LDAT,WRDER	;SETUP FOR HARD ERROR RETURN
1318	003766	013737	017350	001144		MOV	OUTBUF,BUF	;SETUP OUTPUT BUFFER
1319	003774	052737	000003	001132		BIS	#3,FLAG	;SET COUNTER TO 3
1320	004002	004537	007700			JSR	RS,PASEL	;SET UP DATA BUFFERS
1321	004006	005037	001204		LDAT1:	CLR	LOPCNT	;CLEAR ERROR FLAG
1322	004012	104414			LDAT:	ERCLR		;CLEAR RS REG. IF ERROR
1323	004014	004537	006576			JSR	RS,OPDSEL	;SET UP DISK ADDRESS
1324	004020	032737	040000	001132		BIT	#BIT14,FLAG	;TEST FOR WRITE
1325	004026	001462				BEQ	SLH	;TEST FOR READ
1326	004030	012737	000161	001176		MOV	#161,CMD	;WRITE WITH I/E
1327	004036	104416				DKCMD		;DO A WRITE
1328	004040	004737	011622			JSR	PC,WATT	;WAIT FOR INTERRUPT
1329	004044	012737	004012	001010		MOV	#LDAT,LAD	;SETUP SCOPE LOOP
1330	004052	032737	001000	001132		BIT	#BIT9,FLAG	;WAS THERE AN ERROR?
1331	004060	001427				BEQ	WRXBL	;CONT
1332	004062	104430				LOGW		;LOG WRITE ERROR
1333	004064	005237	001204			INC	LOPCNT	;SET ERROR FLAG
1334	004070	022737	000001	001204		CMP	#1,LOPCNT	;IS THIS THE FIRST TIME?
1335	004076	001000				BNE	2\$;NO
1336	004100	004737	014134		2\$:	JSR	PC,PRNT	;TYPE ?
1337	004104	001004				BNE	1\$;NO
1338	004106	104402	000620			TYPE	,DATA	
1339	004112	104402	000630			TYPE	,WRERR	
1340	004116	104044			1\$:	HLT	!DS!DA	;WRITE ERROR
1341	004120	005337	001132			DEC	FLAG	;DEC COUNTER
1342	004124	032737	000003	001132		BIT	#3,FLAG	;DONE YET WITH 3RD TRY?
1343	004132	001327				BNE	LDAT	;NOT 3 TRIES YET? TRY AGAIN
1344	004134	004737	011474			JSR	PC,WTNO	;TYPE CAN NOT WRITE

1345	004140	005737	001204		WRXBL:	TST	LOPCNT		; WAS THERE AN ERROR?
1346	004144	001402				BEQ	WRX1		; NO
1347	004145	004737	011644			JSR	PC, TYPREC		; TYPE RECOVERED
1348	004152	005037	001204		WRX1:	CLR	LOPCNT		; CLEAR ERROR FLAG
1349	004156	104400				SCOPE			
1350	004160	052737	000003	001132		BIS	#3, FLAG		; CLEAR RETRY COUNT
1351	004166	004737	007234			JSR	PC, DISBUF		; SET BUFFER FOR WRITE CHECK
1352	004172	000705				BR	LDA1		
1353	004174	104400			SLH:	SCOPE			
1354	004176	104414			SLH2:	ERCLR			; CLEAR RS REG IF ERRORS
1355	004200	004537	006576			JSR	RS, OPDSEL		; IS THE OPERATOR SELECTING THE TRACK
1356	004204	032737	020000	001132		BIT	#BIT13, FLAG		; TEST FOR WRITE CHECK
1357	004212	001002				BNE	1\$; YES
1358	004214	000137	004520			JMP	ESH1		; NO
1359	004220	013737	017350	001144	1\$:	MOV	OUTBUF, BUF		; SET UP CURRENT ADDRESS
1360	004226	012737	000151	001176		MOV	#151, CMO		; WRITE CHECK WITH I/E
1361	004234	104416				DKCMD			; GO WRITE CHECK
1362	004236	004737	011622			JSR	PC, WATT		; WAIT FOR INTERRUPT
1363	004242	032737	001000	001132	XESH:	BIT	#BIT9, FLAG		; IS THERE AN ERROR?
1364	004250	001505				BEQ	1\$; NO ERROR
1365	004252	005737	001204			TST	LOPCNT		; 1ST ERROR?
1366	004256	001001				BNE	2\$; NO
1367	004260	104434				LOGMC			; YES LOG ERROR
1368	004262	032737	000100	177570	2\$:	BIT	#BIT6, SWR		; TYPE ALL ERRORS?
1369	004270	001007				BNE	3\$; YES
1370	004272	032737	001000	177570		BIT	#BIT9, SWR		; LOOP ON ERROR?
1371	004300	001003				BNE	3\$; YES
1372	004302	005737	001204			TST	LOPCNT		; FIRST ERROR?
1373	004306	001056				BNE	10\$; NO
1374	004310	004737	014134		3\$:	JSR	PC, PRNT		; TYPE OUT?
1375	004314	001052				BNE	4\$; NO
1376	004316	104402	000620			TYPE	, DATA		
1377	004322	104402	000642			TYPE	, WCKERR		
1378	004326	017702	174506			MOV	DRSBA, R2		; GET CORRECT BA
1379	004332	023702	017350			CMP	OUTBUF, R2		; DID A WD GET XFERED?
1380	004336	001406				BEQ	9\$; NO
1381	004340	032737	000400	177570		BIT	#BIT8, SWR		; XFER MODE?
1382	004346	001002				BNE	9\$; YES
1383	004350	162702	000002			SUB	#2, R2		
1384	004354	004737	014134		9\$:	JSR	PC, PRNT		; TYPEOUT ERRORS?
1385	004360	001030				BNE	4\$; NO
1386	004362	005737	001216			TST	MMAVA		; IS MEM MGMT AVAILABLE?
1387	004366	001402				BEQ	7\$; NO

J03

MAINDEC-11-DERSB-A
DERSBB.P11

27-OCT-76

RH11-RS03 DATA AND RELIABILITY TEST
TST2 DATA TEST

MACY11 27(1006) 27-OCT-76 11:08 PAGE 36

1388	004370	004737	007012			JSR	PC,PHYCOV	;YES GET VITURAL ADDR
1389	004374	010237	001226		7S:	MOV	R2,WORK	;GET BA
1390	004400				8S:			
1391	004400	104402	004404			TYPE	,.+2	;.ASCIZ <15><12>*(BA)=""
1392	004414				6S:			
1393	004414	017746	174606			MOV	2WORK,-(6)	;PUT 2WORK ON STACK
1394	004420	104404				TYPE0		;TYPE STACK IN OCTAL
1395	004422	104402	004426			TYPE	,.+2	;ASCIZ "WC="
1396	004434	017746	174376			MOV	2RSWC,-(6)	;PUT 2RSWC ON STACK
1397	004440	104404				TYPE0		;TYPE STACK IN OCTAL
1398	004442	104026			4S:	HLT	!DA!DB!BA	;NOTE: BA REG. = +2 OF ACTUAL MEMORY
1399								;LOC AFTER WORDS HAVE BEEN XFERED
1400	004444	005237	001204		10S:	INC	LOPCNT	;INC ERROR COUNT
1401	004450	022737	000010	001204		CMP	#10,LOPCNT	;10 TRYS YET?
1402	004456	001247				BNE	SLH2	;NO
1403	004460	004737	006750			JSR	PC,NOREC	;TYPE UNRECOVERABLE
1404	004464	005737	001204		1S:	TST	LOPCNT	;ANY ERRORS?
1405	004470	001402				BEQ	5S	;NO
1406	004472	004737	011644			JSR	PC,TYPREC	;TYPE RECOVERED
1407	004476	005037	001204		5S:	CLR	LOPCNT	;CLEAR ERROR COUNTER
1408	004502	104400				SCOPE		
1409	004504	012737	004176	001010		MOV	#SLH2,LAO	;SETUP LOOP ADDRESS
1410	004512	004737	007234			JSR	PC,DISBUF	;SET UP THE DISK BUFFER
1411	004516	000422				BR	SLH2A	
1412	004520	004537	011342		ESH1:	JSR	RS,CLEAR	;CLEAR BUFFER
1413	004524	004537	006576		ESH:	JSR	RS,OPDSEL	;OPERATOR SELECTED DISK ADDRESS?
1414	004530	032737	010000	001132		BIT	#BIT12,FLAG	;TEST FOR READ
1415	004536	001002				BNE	1S	;YES
1416	004540	000137	004754			JMP	MSTR	;NO READ
1417	004544	104400			1S:	SCOPE		
1418	004546	042737	000003	001132		BIC	#3,FLAG	;CLEAR RE-READ COUNT
1419	004554	005037	001204			CLR	LOPCNT	;CLEAR FLAG
1420	004560	000137	004570			JMP	DSKRD	;CONT
1421	004564	000137	004176		SLH2A:	JMP	SLH2	
1422	004570	104414			DSKRD:	ERCLR		;CLEAR RS REG IF ERRORS
1423	004572	012737	000171	001176		MOV	#171,CMD	;READ WITH I/E
1424	004600	104416				OKCMD		;READ
1425	004602	032737	010000	177570		BIT	#10000,SWR	;COMPARE?
1426	004610	001007				BNE	TAG	;NO
1427	004612	032737	000004	001132		BIT	#BIT2,FLAG	;COMPARE?
1428	004620	001003				BNE	TAG	;NO
1429	004622	004537	010444			JSR	RS,COMPARE	;COMPARE
1430	004626	000402				BR	ELH	

K03

MAINDEC-11-DERSB-A
DERSBB.P11 27-OCT-76

RH11-RS03 DATA AND RELIABILITY TEST
TST2 DATA TEST

MACY11 27(1006) 27-OCT-76 11:08 PAGE 37

1431	004630	004737	011622		TAG:	JSR	PC,WATT	:	WAIT FOR INTERRUPT
1432	004634	032737	001000	001132	ELH:	BIT	#BIT9,FLAG	:	IS THERE AN ERROR?
1433	004642	001433				BEQ	ADR0	:	NO
1434	004644	032737	000100	177570		BIT	#BIT6,SWR	:	SOFT ERROR TYPEOUT?
1435	004652	001003				BNE	3\$:	YES
1436	004654	005737	001204			TST	LOPCNT	:	FIRST ERROR?
1437	004660	001011				BNE	2\$:	NO
1438	004662	004737	014134		3\$:	JSR	PC,PRNT	:	TYPEOUT?
1439	004666	001004				BNE	1\$:	NO
1440	004670	104402	000620			TYPE	,DATA		
1441	004674	104402	000657			TYPE	,RDERR		
1442	004700	104432			1\$:	LOGR		:	LOG READ ERROR
1443	004702	104044				HLT	!DS!DA		
1444	004704	104400			2\$:	SCOPE			
1445	004706	005237	001204			INC	LOPCNT	:	COUNT ERRORS
1446	004712	022737	000010	001204		CMP	#10,LOPCNT	:	LAST RETRY?
1447	004720	001323				BNE	DSKRD	:	NO
1448	004722	004737	006750			JSR	PC,NOREC	:	TYPE UNRECOVERABLE
1449	004726	000137	004746		4\$:	JMP	ADR1	:	CONT
1450	004732	104400			ADR0:	SCOPE			
1451	004734	005737	001204			TST	LOPCNT	:	ANY ERRORS?
1452	004740	001402				BEQ	ADR1	:	NO
1453	004742	004737	011644			JSR	PC,TYPREC	:	TYPE RECOVERED
1454	004746	004737	007234		ADR1:	JSR	PC,DISBUF	:	GO SET UP DISK BUFFER.
1455	004752	000500				BR	READCT	:	CONT. READING
1456	004754	005737	001132		MSTR:	TST	FLAG		
1457	004760	100467				BMI	3\$:	OPERATOR SELECTED PATTERN
1458								:	*****
1459								:	*****
1460								:	*****
1461								:	*****
1462								:	*****
1463								:	*****
1464								:	*****
1465								:	*****
1466								:	*****
1467	004762	013746	001134			MOV	WROCT,-(SP)		
1468	004766	000316				SWAB	(SP)		
1469	004770	042716	177400			BIC	#177400,(SP)		
1470	004774	006216				ASR	(SP)		
1471	004776	006216				ASR	(SP)		
1472	005000	006216				ASR	(SP)		
1473	005002	006216				ASR	(SP)		
1474	005004	061637	001026			ADD	(SP),TEMP3		
1475	005010	023737	001022	001026		CMP	TEMP1,TEMP3		
1476	005016	003424				BLE	5\$		
1477	005020	061637	001026			ADD	(SP),TEMP3		
1478	005024	023737	001022	001026		CMP	TEMP1,TEMP3		
1479	005032	003416				BLE	4\$		
1480	005034	162637	001026			SUB	(SP)+,TEMP3		
1481	005040	005037	017350			CLR	OUTBUF		
1482	005044	013746	001026			MOV	TEMP3,-(SP)		
1483	005050	062737	020000	017350	6\$:	ADD	#20000,OUTBUF		
1484	005056	005316				DEC	(SP)		
1485	005060	001373				BNE	6\$		
1486	005062	005726				TST	(SP)+		

L03

1487	005054	000137	003726			JMP	LDAT3		
1488	005070				4S:				
1489	005070	005726			5S:	TST	(SP)+		
1490	005072	062737	000002	001142		ADD	#2,PATNU	; INC PATTERN INDEX	
1491	005100	022737	000044	001142		CMP	#44,PATNU		
1492	005106	071402				BEQ	.+6		
1493	005110	000137	003710			JMP	DATAT	; NOT LAST PATTERN EXIT	
1494	005114	005037	001142			CLR	PATNU	; LAST PATTERN EXIT	
1495	005120	032737	000002	177570		BIT	#BIT1,SWR	; DATA TEST ONLY?	
1496	005126	001006				BNE	2S	; YES	
1497	005130	032737	002000	001132		BIT	#BIT10,FLAG	; DATA TEST ONLY?	
1498	005136	001404				BEQ	1S	; NO	
1499	005140	005137	001120		3S:	COM	AOB1	; ALTERNATE PORTS	
1500	005144	000137	006204		2S:	JMP	EXTPPR	; LOOP	
1501	005150	000137	005166		1S:	JMP	RANEX	; DO NEXT TEST	
1502									
1503	005154	000137	004524		READCT:	JMP	ESH	; CONT. READING	
1504	005160	052737	100000	001222	RANEL:	BIS	#BIT15,FLAG3	; SET LOOP IN RANDOM TEST GOT	
1505								; HERE BY STARTING AT LOC 264	

M03

MAINDEC-11-DERSB-A
DERSB8.P11 27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST
TST2 DATA TEST

MACY11 27(1006) 27-OCT-76 11:08 PAGE 39

1506 005166
1507
1508
1509
1510
1511
1512
1513 005166 104400
1514
1515
1516
1517
1518
1519
1520
1521
1522 005170 005037 001024
1523 005174 052737 000400 001132
1524 005202 012737 020000 017350
1525 005210 013737 017350 001122
1526 005216 005737 001216
1527 005222 001402
1528 005224 005037 177572
1529 005230 012737 000042 001142
1530 005236 104426
1531 005240 012737 176030 001162
1532 005246 012737 006024 001156
1533 005254 004737 007160
1534 005260 012777 000340 173602
1535 005266 012737 005406 001010
1536 005274 012737 000001 001226
1537 005302 013701 017350
1538 005306 004537 010040
1539 005312 017737 012032 001140
1540 005320 042737 170000 001140
1541 005326 052737 000003 001132
1542 005334 012737 000200 001134
1543 005342 005737 001172
1544 005346 001003
1545 005350 012737 000100 001134
1546 005356 013737 001134 001226
1547 005364 013701 017350
1548 005370 004537 010040
1549 005374 013737 017350 001144
1550 005402 005037 001204

RANEX: ;RANDOM ADDRESS DATA TEST
;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
XYZ*****?*****
*****
XYZ*****?*****
*****
XYZ*****?*****
*****
```

```
CLR      TEMP2          ;NOMORE A TEST 2
2$:      BIS      #BIT8,FLAG ;SET TEST FLAG
         MOV      #20000,OUTBUF ;GET STARTING ADDR OF BUF
         MOV      OUTBUF,VADDR ;SAVE BUFFER ADDR
         TST      MMAVA       ;MEM MGMT AVAILABLE?
         BEQ     1$          ;NO
         CLR     2$SR0       ;TURN IT OFF
1$:      MOV      #42,PATNU   ;DO RANDOM COMPARE
         CLR     2$SR0
         CLRDV
         MOV     #-1000.,PASSC ;SET UP PASS COUNT
         MOV     #RWRED,ARDR   ;SET UP FOR HARD ERROR
1533:    JSR     PC,VECTR     ;SETUP INTERRUPT VECTOR
         MOV     #340,2$STATUS
WRLG1:   MOV     #WRERR,LAD   ;SETUP LOOP ADDRESS
         MOV     #1,WORK      ;SET UP RANDOM GENERATOR WORD
         MOV     OUTBUF,R1
         JSR     R5,RANDOM    ;GENERATE RANDOM DATA
         MOV     2$OUTBUF,DMA ;SET UP DISK ADDRESS
         BIC     #170000,DMA
         BIS     #3,FLAG
         MOV     #200,WRDCT   ;SET COUNTER
1542:    MOV     R504
         TST     R504?
         BNE     2$          ;YES
         MOV     #100,WRDCT   ;SET UP WORD COUNT =1SECTOR
2$:      MOV     WRDCT,WORK   ;GENERATE RANDOM BUFFER
         MOV     OUTBUF,R1
         JSR     R5,RANDOM
         MOV     OUTBUF,BUF
         CLR     LOPCNT
         ;SET UP OUTPUT BUFFER
         ;CLR ERROR FLAG
```


1574	005536	005737	001204		WRRCK1:	TST	LOPCNT		: ANY ERRORS?
1575	005532	001402				BEQ	15		: NO
1576	005534	004737	011644			JSR	PC, TYPREC		: TYPE RECOVERED
1577	005540	104400			15:	SCOPE			
1578	005542	005C37	001204			CLR	LOPCNT		: CLEAR LOOP COUNT
1579	005546	104414			WRRCK:	ERCLR			: CLEAR RS REG IF ERRORS
1580	005550	012737	000151	001176		MOV	#151, CMD		: WRITE CHECK WITH I/E
1581	005556	104416				DKCMD			: WRITE CHECK
1582	005560	004737	011622			JSR	PC, WATT		: WAIT FOR INTERRUPT
1583	005564	032737	001000	001132	45:	BIT	#BIT3, FLAG		: ERROR?
1584	005572	001453				BEQ	15		: NO
1585	005574	032737	000100	177570		BIT	#BIT6, SWR		: TYPE ALL RETRYS?
1586	005580	001003				BNE	25		: YES
1587	005584	005737	001204			TST	LOPCNT		: FIRST ERROR?
1588	005590	001030				BNE	55		: NO
1589	005592	104434			25:	LOGWC			: LOG WRITE CK
1590	005594	004737	014134			JSR	PC, PRNT		: TIMEOUT?
1591	005620	001052				BNE	65		: NO
1592	005622	104402	000670			TYPE	, RANDM		
1593	005626	104402	000642			TYPE	, WCKERR		
1594	005632	017737	173202	001226		MOV	#RSBA, WORK		: GET CORRECT BA
1595	005640	162737	000002	001226		SUB	#2, WORK		
1596	005646	104402	005652			TYPE	, +2		: .ASCIZ (15)<(12)<(BA)=""
1597	005662	017746	173340			MOV	#WORK, -(6)		: PUT #WORK ON STACK
1598	005666	104404				TYPE0			: TYPE STACK IN OCTAL
1599	005670	104026				HLT	!DB!DA!BA		: BA=MEMORY LOC +2 OF ACTUAL WORD
1600	005672	005237	001204		55:	INC	LOPCNT		: INC RETRY COUNT
1601	005676	022737	000010	001204		CMP	#10, LOPCNT		: LAST ONE YET?
1602	005704	001320				BNE	WRRCK		: NO
1603	005706	104432	006701			TYPE	, UNRECO		
1604	005712	005337	001204			CLR	LOPCNT		: CLEAR LOPCNT
1605	005715	000137	006126			JMP	EXRAX		: GET NEW NUMBER
1606	005722	005737	001204		15:	TST	LOPCNT		: ANY ERRORS?
1607	005726	001407				BEQ	65		: NO
1608	005730	104402	000702			TYPE	, RECOV		
1609	005734	013746	001204			MOV	LOPCNT, -(6)		: GET NUMBER
1610	005740	104406				TYPES			: TYPE IT
1611	005742	104402	000757			TYPE	, CALFLF		
1612	005746	104400			65:	SCOPE			
1613	005750	052737	000003	001132		BIS	#3, FLAG		: SET COUNTER

1614	005756	104400				SCOPE		
1615	005760	005037	001204			CLR	LOPCNT	: CLEAR COUNTER
1616	005764	004537	011342			JSR	RS, CLEAR	: CLEAR BUFFER
1617	005770	104414			RREAD:	ERCLR		: CLEAR RS REG IF ERRORS
1618	005772	012737	000171	001176		MOV	#171, CMD	: READ WITH I/E
1619	006000	104416				DKCMD		: READ
1620	006002	032737	010000	177570		BIT	#BIT12, SWR	: COMPARE ?
1621	006010	001003				BNE	TAG1	: NO
1622	006012	004537	010444			JSR	RS, COMPARE	: YES
1623	006016	000402				BR	RWRED	: CONT
1624	006020	004737	011622		TAG1:	JSR	PC, WAIT	: WAIT FOR INTERRUPT
1625	006024	032737	001000	001132	RWRED:	BIT	#BIT9, FLAG	: IS THERE AN ERROR?
1626	006032	001435				BEQ	EXRAX	: NO
1627	006034	104432				LOGR		: LOG READ ERR
1628	006036	032737	000100	177570	1S:	BIT	#BIT6, SWR	: TYPE ALL ERRORS?
1629	006044	001016				BNE	2S	: YES
1630	006046	032737	001000	177570		BIT	#BIT9, SWR	: LOOP ON ERROR?
1631	006054	001012				BNE	2S	: YES
1632	006056	005737	001204			TST	LOPCNT	: FIRST ERROR?
1633	006062	001010				BNE	3S	: NO
1634	006064	004737	014134			JSR	PC, PRNT	: TYPEOUT?
1635	006070	001004				BNE	2S	: NO
1636	006072	104402	000670			TYPE	, RANDM	
1637	006076	104402	000657			TYPE	, RDERR	
1638	006102	104036			2S:	HLT	!DB!DA	
1639	006104	104400			3S:	SCOPE		
1640	006106	005237	001204			INC	LOPCNT	: UPDATE COUNTER
1641	006112	022737	000010	001204		CMP	#10, LOPCNT	: LAST TRY YET?
1642	006120	001323				BNE	RREAD	: RETRY
1643	006122	004737	006750			JSR	PC, NOREC	: TYPE UNRECOVERABLE
1644	006126	005737	001204		EXRAX:	TST	LOPCNT	: ANY ERRORS?
1645	006132	001402				BEQ	EXRXX	: NO
1646	006134	004737	011644			JSR	PC, TYPREC	: TYPE RECOVERED
1647	006140	104400			EXRXX:	SCOPE		
1648	006142	005237	001162			INC	PASSC	: +1 PASS COUNT
1649	006146	001402				BEQ	1S	: IS TEST DONE?
1650	006150	000137	005266			JMP	WRLG1	: NO
1651	006154	005037	001142		1S:	CLR	PATNU	: END OF TEST
1652	006160	042737	000400	001132		BIC	#BIT8, FLAG	: CLEAR TEST FLAG
1653	006166	032737	100000	001222		BIT	#BIT15, FLAG3	: LOOP ON THIS TEST?
1654	006174	001402				BEQ	.+6	: NO
1655	006176	000137	005166			JMP	RANEX	: YES

```

1656 ;CHECK FOR MULTI DISK MODE
1657 ;IF IN MULTI DISK MODE REPORT "END"
1658 ;IF LAST DISK ON SYSTEM HAS BEEN EXERCISED.
1659
1660 ;*****%*****
1661 ;TEST 4 TEST FOR MULTI DISK MODE
1662 ;*****%*****
1663 006202 104400 TST4: SC0F2
1664 006204 005037 001140 EXTPPR: CLR DMA
1665 006210 005037 001024 CLR TEMP2
1666 006214 104426 CLR0V ;INIT DRIVE
1667 006216 032737 004000 001132 BIT #BIT11,FLAG ;ARE WE IN MULTI DISK MODE
1668 006224 001404 BEQ EXTPP ;NO REPORT "END"
1669 006226 004737 002130 JSR PC,TRYNX ;YES TEST FOR ALL DRIVES
1670 006232 000137 003242 JMP EX4FLG ;RESTART TESTING OF DRIVES
1671 006236 004737 002204 EXTPP: JSR PC,DONEE ;GET PASS COUNT
1672
1673 ;THIS ROUTINE CLEARS THE DRIVE
1674 ;REGISTERS IF THERE WAS AN ERROR
1675
1676 006242 032737 001000 001132 .ERCLR: BIT #BIT9,FLAG ;ANY ERRORS?
1677 006250 001404 BEQ IS ;NO
1678 006252 104426 CLR0V ;CLEAR ALL ERRORS
1679 006254 042737 001000 001132 BIC #BIT9,FLAG ;CLEAR ERROR FLAG
1680 006262 000002 IS: RTI ;EXIT
1681
1682 ;ENTER DISK HANDLER BY THE TRAP INSTRUCTION
1683 ;ARGUMENT TO TRAP INSTRUCTION IS TWO ORDER
1684 ;BYTE OF THE CONTROL REGISTER.
1685
1686 006264 013777 001140 172550 .DKCMD: MOV DMA,DRSDA ;LOAD DISK ADD
1687 ;*****%*****
1688 ;*****%*****
1689 ;*****%*****
1690 ;*****%*****
1691 ;*****%*****
1692 ;*****%*****
1693 ;*****%*****
1694 ;*****%*****
1695 006272 005737 001024 TST TEMP2
1696 006276 001422 BEQ 5$
1697 006300 013746 001026 MOV TEMP3, -(6)
1698 006304 005077 172552 CLR DRSDAE
1699 006310 005037 001030 CLR TEMP4
1700 006314 062737 020000 001030 6$: ADD #20000,TEMP4
1701 006322 013737 001030 001144 MOV TEMP4,BUF
1702 006330 103002 BCC 7$
1703 006332 005277 172524 INC DRSDAE
1704 006336 005316 7$: DEC (6)
1705 006340 001365 BNE 6$
1706 006342 005726 TST (6)+
1707 006344 005037 001202 5$: CLR INTFLG ;CLEAR INTERRUPT FLAG
1708 006350 013777 001144 4$: MOV BUF,DRSBA ;LOAD (DMA) BUSS ADDRESS
1709 006356 013702 001134 MOV WROCT,R2 ;GET NEGATIVE
1710 006362 005402 NEG R2 ;WORD COUNT
1711 006364 010277 172446 MOV R2,DRSWC ;LOAD WC

```

E04

1712	006370	032737	000400	001132	BIT	#BIT8,FLAG	:RANDOM TEST?
1713	006376	001021			BNE	1\$:YES A PORT ONLY WITH NO MEM MGMT
1714	006400	005737	001216		TST	MMAVA	:MEM MGMT AVAILIABLE?
1715	006404	001416			BEQ	1\$:NO
1716	006406	032737	000040	001126	BIT	#BIT5,FLAG2	:SET A17 IN RSCSI
1717	006414	001403			BEQ	3\$:NO
1718	006416	052737	001000	001176	BIS	#BIT9,CMD	:YES

F04

```

1719 006424 032737 000020 001126 3$: BIT #BIT4,FLAG2 ;SET A16?
1720 006432 001403 BEQ 1$ ;NO
1721 006434 052737 000400 001176 BIS #BIT8,CMD ;YES
1722 006442 113777 001176 172362 1$: MOV8 CMD,#RSCS1 ;LOAD FUNCTION REG.
1723 006450 000002 RTI ;RETURN FROM TRAP
1724
1725 ;PH11 DISK INTERRUPT HANDLER
1726 ;ROUTINE CONTINUES ON ERRORS
1727
1728 006452 042737 001000 001132 DKINT: BIC #BIT9,FLAG ;CLEAR ERROR BIT
1729 006460 005777 172346 TST #RSCS1 ;TEST FOR ERROR
1730 006464 100401 BMI 2$
1731 006466 000425 BR INTEXT ;JUMP IF NO ERRORS
1732 006470 017702 172336 2$: MOV #RSCS1,R2 ;GET CONTENTS OF CS1
1733 006474 042702 037777 BIC #37777,R2 ;CLEAR ALL BUT SC AND TRE
1734 006500 022702 140000 CMP #140000,R2 ;IS SC AND TRE BOTH SET?
1735 006504 001413 BEQ TRUERR ;YES THERE IS SOME KIND OF XFER ERROR
1736 006506 032777 100000 172330 BIT #100000,#RSDS ;IS THE ATA BIT SET?
1737 006514 001007 BNE TRUERR ;YES
1738 006516 104140 HLT !AS!DS ;WRONG UNIT INTERRUPTED
1739 ;IF YOU HAVE JUST POWERED UP A DRIVE OR
1740 ;ARE RUNNING THE POWER FAIL TEST,
1741 ;INTERRUPTS WILL OCCUR FROM DRIVES OTHER
1742 ;THAN THE UNIT UNDER TEST. IF THIS TYPEOUT
1743 ;SHOWS NO ERRORS IN THE REGISTERS OF THE DRIVE
1744 ;UNDER TEST, THAT DRIVE IS OK
1745 006520 012777 177777 172322 1$: MOV #-1,#RSDS ;CLEAR ALL ATA BITS
1746 006526 013716 001156 MOV #RSDS,(SP) ;GET RETURN ADD.
1747 006532 000002 RTI ;RETRY
1748 006534 052737 001000 001132 TRUERR: BIS #BIT9,FLAG ;SET ERROR BIT
1749 006542 032737 004000 177570 INTEXT: BIT #BIT11,SWR ;HALT ON COMPLETION FLAG
1750 006550 001401 BEQ .+4
1751 006552 000000 HALT
1752 006554 032737 002000 001126 BIT #BIT10,FLAG2 ;YES BIT 11 SET IN SWR HALT
1753 006562 001402 BEQ 1$ ;WAIT IN BACKGROUND TEST?
1754 006564 012716 012306 2$: MOV #NPRRET,(SP) ;NO
1755 006570 010637 001202 1$: MOV SP,INTFLG ;MODIFY RETURN ADD.
1756 006574 000002 RTI ;SET INT FLG
1757 ;ROUTINE TO SET UP TRACK # FROM OPTION ;EXIT
1758 ;ENTER FROM JSR R5, OPDSEL
1759
1760 006576 032737 000040 001132 OPDSEL: BIT #BIT5,FLAG ;OPTIONAL DMA?
1761 006604 001403 BEQ 1$ ;NO
1762 006606 013737 001146 001140 MOV TDMA,DMA ;GET OPT. DMA
1763 006614 000205 1$: RTS R5 ;EXIT
  
```

G04

1764	006616	005015	052123	051101	STABUF: .ASCIZ	<15><12>"STARTING 4K BANK #"
1765	006624	044524	043516	032040		
1766	006632	020113	040502	045516		
1767	006640	021440	000			
1768						
1769	006643	015	044012	053517	BUFSIZ: .ASCIZ	<15><12>"HOW MANY 4K BANKS? (OCTAL) "
1770	006650	046440	047101	020131		
1771	006656	045464	041040	047101		
1772	006664	051513	020077	047450		
1773	006672	052103	046101	020051		
1774	006700	000				
1775						
1776	006701	015	052412	051116	UNRECO: .ASCIZ	<15><12>"UNRECOVERABLE"<15><12><12>
1777	006706	041505	053117	051105		
1778	006714	041101	042514	005015		
1779	006722	000012				
1780						
1781	006724	005015	047125	041101	NOWRIT: .ASCIZ	<15><12>"UNABLE TO WRITE"<15><12>
1782	006732	042514	052040	020117		
1783	006740	051127	052111	006505		
1784	006746	000012				
1785						
1786					.EVEN	
1787						
1788	006750	004737	014134		NOREC: JSR	PC,PRNT ;TYPEOUT?
1789	006754	001002			BNE	IS ;NO
1790	006756	104402	006701		TYPE	,UNRECO
1791	006762	005037	001204		IS: CLR	LOPCNT ;CLEAR LOOP COUNTER
1792	006766	000207			RTS	PC
1793						
1794	006770	052737	040000	001126	FNDTYP: BIS	#BIT14,FLAG2 ;SET CHECK DRIVE TYPE FLAG
1795	006776	004737	001762		JSR	PC,UNTP ;CHECK DRIVE TYPE FLAG
1796	007002	042737	040000	001126	BIC	#BIT14,FLAG2 ;CLEAR DRIVE TYPE FLAG
1797	007010	000207			RTS	PC

H04

```
1798 ;ROUTINE TO CALCULATE VITURAL ADDR
1799
1800 PHYCOV: SWAB R2 ;CALCULATE FROM PHYSICAL ADDR
1801 JSR PC,RRR2
1802 ROR R2
1803 BIC #177770,R2 ;GET REG #
1804 BIT #BIT8,DRSCSI ;IS A16 SET?
1805 BEQ 1$ NO
1806 BIS #BIT3,R2 YES
1807 BIT #BIT9,DRSCSI ;IS A17 SET?
1808 BEQ 2$ NO
1809 BIS #BIT4,R2 YES
1810 MOV STAMEM,WORK1 ;GET BANK # FOR -A- PORT
1811 *****
1812 *****
1813 *****
1814 *****
1815 *****
1816 *****
1817 *****
1818 TST TEMP2
1819 BEQ 3$
1820 MOV TEMP3,WORK1
1821 SUB WORK1,R2 ;GET STARTING BANK #
1822 ADD #1,R2 ;GET OFFSET FOR REG #
1823 SWAB R2 ;GET BANK # INTO
1824 ROL R2 ;UPPER BITS
1825 ROL R2
1826 ROL R2
1827 ROL R2
1828 ROL R2
1829 MOV DRSCBA,WORK1 ;GET OFFSET FOR ADDR IF ANY
1830 SUB #2,WORK1 ;CORRECT IT
1831 BIC #160000,WORK1 ;CLEAR JUNK
1832 BIS R2,WORK1 ;GET REG NO
1833 MOV WORK1,R2
1834 RTS PC
1835
1836 VECTR: MOV #DKINT,DRSVEC ;SETUP INTERRUPT VECTORS
1837 MOV PRIORITY,PS ;PRIORITY 4
1838 RTS PC
1839
1840 ;THIS ROUTINE IS USED FOR DELAYING THE START OF THIS PROGRAM
1841 ;IF POWER FAILED DURING TESTING. THIS WILL GIVE THE DRIVES TIME TO GET UP
1842 ;TO SPEED. THE DELAY WILL BE ABOUT 3-5 MINUTES DEPENDING UPON THE PROCESSOR
1843
1844 TIMUP: MOV #177777,WORK
1845 1$: MOV #177777,WORK1
1846 2$: NOP
1847 DEC WORK1
1848 BNE 2$
1849 DEC WORK
1850 BNE 1$
1851 JMP ADTST
1852 ;ROUTINE TO SETUP DISK BUFFERS
1853 ;ADD WORD COUNT TO STARTING DISK ADDRESSES
```

```

1854 ;COMPARE CALCULATED ADDRESS TO TERMINATING ADDRESS
1855
1856 007234 032737 000040 001132 DISBUF: BIT #BITS,FLAG ;DID OPERATOR SELECT PATTERNS
1857 007242 001402 BEQ 1$ ;NO
1858 007244 000137 007410 JMP BUFEXIT ;YES
1859 007250 004737 007600 1$: JSR PC,BLSZ ;DEFINE BLOCK SIZE
1860 007254 013737 001160 001230 MOV BLOCK,WORK1
1861 007262 005237 001140 INCSEC: INC DMA ;+1 SECTOR COUNT
1862 007266 022737 010000 001140 CMP #10000,DMA ;DONE YET?
1863 007274 001445 BEQ BUFEXIT ;YES
1864 007276 005337 001160 DEC BLOCK ;-1 FROM BLOCK COUNT
1865 007302 001401 BEQ COMDAR ;CMP DMA TO RSDA
1866 007304 000766 BR INCSEC ;RECYCLE
1867 007306 032737 001000 001132 COMDAR: BIT #BITS,FLAG ;ANY ERRORS?
1868 007314 001401 BEQ 1$ ;NO ERRORS DO COMPARE ON RSDA
1869 007316 000207 RTS PC ;ERRORS DO NOT COMPARE RSDA
1870 007320 023777 001140 171514 1$: CMP DMA,RSDA ;COMPARE RSDA WITH DMA
1871 007326 001425 BEQ CMDAE ;SHOULD BE EQUAL
1872 007330 104432 LOGR ;AFTER TRANSFER RSDA AND DMA SHOULD BE =
1873 ;IF NOT, RSDA IS NOT CORRECT. DMA CONTAINS
1874 ;WHAT RSDA SHOULD =
1875 007332 013701 001140 MOV DMA,GOOD ;GET DMA FOR CORRECT ANS IN GOOD
1876 007336 017700 171500 MOV @RSDA,BAD ;GET RSDA INTO BAD
1877 007342 104000 HLT ;RSDA=BAD DMA=GOOD SEE COMMENTS 7 LINES ABOVE
1878 007344 004737 014134 JSR PC,PRNT ;TYPEOUT?
1879 007350 001014 BNE CMDAE ;NO
1880 007352 011637 001226 MOV (SP),WORK ;GET TEST PC FROM WHERE IT CAME
1881 007356 104402 007362 TYPE ;ASCIZ " TST PC="
1882 007374 013746 001226 MOV WORK,-(6) ;PUT WORK ON STACK
1883 007400 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
1884 007402 105737 001132 CMDAE: TSTB FLAG ;LAST DISK BUFFER?
1885 007406 100032 BPL BUFINX ;NO
    
```


J04

1886	007410	005037	001140		BUFEXIT: CLR	DMA	; CLEAR ADDRESS BITS LAST DISK BUFFER
1887	007414	062716	000002		ADD	#2, (6)	; INC STOCK POINTER
1888	007420	042737	000200	001132	AKH: BIC	#200, FLAG	; CLEAR LAST DISK BUFFER FLAG
1889	007426	032737	000400	001132	BIT	#BIT8, FLAG	; RANDOM TEST OR ADDR TEST?
1890	007434	001404			BEQ	1\$; NO
1891	007436	013737	001150	001134	2\$: MOV	SWRDC, WRDCT	
1892	007444	000454			BR	EXTDR	; EXIT
1893	007446	032737	000100	001132	1\$: BIT	#BIT6, FLAG	; MULTI PORT?
1894	007454	001770			BEQ	2\$; NO
1895	007456	005737	001120		TST	A0B1	; A OR B PORT?
1896	007462	001765			BEQ	2\$; A PORT
1897	007464	013737	001116	001134	MOV	WDCTB, WRDCT	; B PORT
1898	007472	000441			BR	EXTDR	; GET OUT
1899	007474	005037	001232		BUFINX: CLR	WORK2	; CLEAR WORK2 FOR BLOCK COUNTER
1900	007500	013702	001140		MOV	DMA, R2	; PUT WORKING DISK ADDR INTO WORK
1901	007504	005237	001232		XINCSEC: INC	WORK2	; INCREMENT BLOCK COUNT
1902	007510	022702	007777		CMP	#7777, R2	; CMP FOR LAST SECTOR
1903	007514	001405			BEQ	XINCSUR	; +1 SURFACE LAST SECTOR BRANCH
1904	007516	005202			INC	R2	; INC DMA
1905	007520	005337	001230		DEC	WORK1	; DEC BLOCK COUNT
1906	007524	001367			BNE	XINCSEC	; FILLED STANDARD BUFFER YET?
1907	007526	000734			BR	AKH	; WILL TAKE STANDARD SIZE WORD COUNT
1908	007530	013746	001232		XINCSUR: MOV	WORK2, -(SP)	; SETTING UP BLOCK COUNT
1909	007534	000241			CLC		; FOR NON STANDARD BUFFER SIZE
1910	007536	006116			ROL	(SP)	
1911	007540	006116			ROL	(SP)	
1912	007542	006116			ROL	(SP)	
1913	007544	006116			ROL	(SP)	
1914	007546	006116			ROL	(SP)	
1915	007550	006116			ROL	(SP)	
1916	007552	012637	001134		MOV	(SP)+, WRDCT	
1917	007556	005737	001172		TST	RS040†	; RS04?
1918	007562	001402			BEQ	1\$; NO
1919	007564	006137	001134		ROL	WRDCT	; YES
1920	007570	052737	000200	001132	1\$: BIS	#200, FLAG	; SET LAST DISK BUFFER FLAG
1921	007576	000207			EXTDR: RTS	PC	; EXIT

K04

```

1922 ;THIS ROUTINE CONVERTS A WORD COUNT TO A BLOCK COUNT
1923
1924 007600 012737 000177 001160 BLSZ: MOV #177,BLOCK ;SETUP FOR RSD4
1925 007606 005737 001172 TST RSD4DT ;RSD4?
1926 007612 001003 BNE 2$ ;YES
1927 007614 012737 000077 001160 1$: MOV #77,BLOCK ;PUT SECTOR SIZE INTO BLOCK
1928 007622 013702 001134 2$: MOV WRDCT,R2 ;FETCH WORD COUNT
1929 007626 033702 001160 BIT BLOCK,R2 ;ARE THEY EQUAL?
1930 007632 001406 BEQ RORBLK ;YES
1931 007634 043702 001160 BIC BLOCK,R2 ;SET UP BLOCK OVERFLOW
1932 007640 005237 001160 INC BLOCK
1933 007644 063702 001160 ADD BLOCK,R2
1934 007650 000241 RORBLK: CLC
1935 007652 006002 ROR R2
1936 007654 006002 ROR R2
1937 007656 004737 011610 JSR PC,RRR2
1938 007662 005737 001172 TST RSD4DT ;RSD4?
1939 007666 001401 BEQ 1$ ;NO
1940 007670 006002 ROR R2 ;YES
1941 007672 010237 001160 1$: MOV R2,BLOCK ;BLOCK COUNT
1942 007676 000207 RTS PC ;EXIT
1943
1944 ;ROUTINE TO SELECT DATA PATTERNS FOR TEST
1945
1946 ;ENTER FROM JSR R5,PASEL
1947
1948 007700 012737 010374 000004 PASEL: MOV #MEM,2#4 ;SETUP TRAP
1949 007706 012737 000340 000006 MOV #340,2#6 ;VECTOR
1950 007714 013700 001142 MOV PATNU,R0 ;SET UP PATTERN NUMBER
1951 007720 010003 MOV R0,R3 ;GET PATTERN #
1952 007722 000241 CLC ;MAKE IT =
1953 007724 006003 ROR R3 ;TO PATTERN # IN LISTING
1954 007726 010337 177570 MOV R3,2#DISPLAY ;DISPLAY PATTERN #
1955 007732 013737 001134 001226 MOV WRDCT,WORK ;SET UP WORK
1956 007740 013701 001122 MOV VADDR,R1 ;LOC. OF OUTBUFFER
1957 007744 022700 000042 1$: CMP #42,R0 ;TEST FOR RANDOM DATA NUMBER
1958 007750 001433 BEQ RANDOM ;GO GENERATE RANDOM DATA
1959 007752 032737 000004 001132 BIT #BIT2,FLAG ;MAX TST?
1960 007760 001404 BEQ 2$ ;NO
1961 007762 015037 000300 017350 MOV PATO(0),OUTBUF ;GET PATTERN
1962 007770 000205 RTS R5
1963 007772 016000 000300 2$: MOV PATO(0),R0
1964 ;XYZ*****?*****
1965 ;*****
1966 ;*****
1967 ;XYZ*****?*****
1968 ;*****
1969 ;*****
1970 ;XYZ*****?*****
1971 ;*****
1972 007776 004737 017414 JSR PC,MMPSET ;PRESET MEMORY MANAGEMENT REG
1973 010002 005737 001024 TST TEMP2 ;IS IT A TEST 2
1974 010006 001756 BEQ 1$ ;BRANCH IF NOT TEST 2
1975 010010 004537 017352 FILDAT: JSR R5,MMUSE ;DO NEXT INSTRUCTION WITH MM
1976 010014 010021 1$: MOV R0,(1)+ ;FILL BUFFER
1977 010016 005337 001226 DEC WORK ;DEC. WORK COUNT

```

L04

MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 51
DERSBB.P11 27-OCT-76 11:03 TST4 TEST FOR MULTI DISK MODE

1978	010022	001372				BNE	FILDAT	;LOAD NEXT WORD
1979	010024	012737	000006	000004	PASEX:	MOV	#6, #4	;RESTORE
1980	010032	005037	000006			CLR	#6	;TRAP
1981	010036	000205				RTS	R5	;BUFFER FULL

M04

;RANDOM DATA GENERATOR SUBROUTINE

1982
1983
1984 010040 013737 010234 010240
1985 010046 013737 010236 010242
1986
1987
1988
1989
1990
1991
1992
1993
1994 010054 004737 017414
1995 010060 013700 010234
1996 010064 013704 010236
1997 010070 012703 000007
1998 010074 005002
1999 010076 006300
2000 010100 006104
2001 010102 006102
2002 010104 005303
2003 010106 001373
2004 010110 063700 010234
2005 010114 005504
2006 010116 063704 010236
2007 010122 005502
2008 010124 062700 001057
2009 010130 005504
2010 010132 005502
2011 010134 062704 047401
2012 010140 005502
2013 010142 062702 000006
2014 010146 062700 000002
2015 010152 005504
2016 010154 010037 010234
2017
2018
2019
2020
2021
2022
2023
2024
2025 010160 005737 001024
2026 010164 001402
2027 010166 004537 017352
2028 010172 010021
2029 010174 005337 001226
2030 010200 001413
2031 010202 010437 010236
2032
2033
2034
2035
2036
2037

```
RANDOM: MOV LONUM, LOSAV  
MOV HINUM, HISAV  
:XYZ*****?  
:*****  
:*****  
:XYZ*****?  
:*****  
:*****  
:XYZ*****?  
:*****  
:*****  
RAND1: JSR PC, MMPSET ;PRESET MEMORY MANAGEMENT REG  
MOV LONUM, R0 ;SET UP R0 WITH 5 DIGITS LOW  
MOV HINUM, R4 ;SET UP R1 WITH 5 DIGITS HIGH  
MOV #7, R3 ;SET UP SHIFT COUNT  
SHIFT: CLR R2 ;CLEAR R2  
ASL R0 ;SHIFT R0 LEFT AND  
ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO  
ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2  
DEC R3 ;DECREMENT R3  
BNE SHIFT ;CONTINUE SHIFT LOOP  
ADD LONUM, R0 ;ADDN IN NUMBER TO MAKE X 129  
ADC R4 ;PROPOGATE CARRY  
ADD HINUM, R4 ;ADDN IN NUMBER TO MAKE X 129  
ADC R2 ;PROPOGATE CARRY  
ADD #1057, R0 ;ADDN LOW CONSTANT  
ADC R4 ;PROPOGATE CARRIES  
ADC R2 ;PROPOGATE AGAIN  
ADD #47401, R4 ;ADDN HIGH CONSTANT  
ADC R2 ;PROPOGATE CARRY  
ADD #6, R2 ;ADDN HIGHEST CONSTANT  
ADD #2, R0 ;REPRIME R0 WITH HIGH DIGIT  
ADC R4 ;PROPOGATE CARRY  
MOV R0, LONUM ;PUT R0 BACK IN LONUM  
:XYZ*****?  
:*****  
:*****  
:XYZ*****?  
:*****  
:*****  
:XYZ*****?  
:*****  
:*****  
TST TEMP2 ;IS IT A TEST 2  
BEQ IS ;BRANCH IF NOT TEST 2  
JSR R5, MMUSE ;DO NEXT INSTRUCTION WITH MM  
IS: MOV R0, (1)+ ;HOLD LONUM FOR PROGRAM  
DEC WORK  
BEQ EXGEN  
MOV R4, HINUM ;PUT R1 BACK IN HINUM  
:XYZ*****?  
:*****  
:*****  
:XYZ*****?  
:*****  
:*****
```

N04

```

2038 ;XYZ*****?*****
2039 ;*****
2040 010206 005737 001024 TST TEMP2 ;IS IT A TEST 2
2041 010212 001402 BEQ 25 ;BRANCH IF NOT TEST 2
2042 010214 004537 017352 JSR R5,MMUSE ;DO NEXT INSTRUCTION WITH MM
2043 010220 010421 25: MOV R4,(1)+ ;HOLD HINUM FOR PROGRAM
2044 010222 005337 001226 DEC WORK
2045 010226 001314 BNE RAND1
2046 010230 000137 010024 EXGEN: JMP PASEX ;RETURN TO PROGRAM
2047 010234 000000 LONUM: 0
2048 010236 000000 HINUM: 0
2049 010240 000000 LOSAV: 0
2050 010242 000000 HISAV: 0
2051
2052 010244 013737 001220 001150 RESTOR: MOV SAVWC,SWDCT ;RESTORE ORIGINAL
2053 010252 013737 001150 001134 MOV SWDCT,WRDCT ;WORD COUNT
2054 010260 000205 RTS R5

```

B05

2055
2056
2057
2058 010262 013700 010370
2059 010266 013704 010372
2060 010272 012703 000007
2061 010276 005002
2062 010300 006300
2063 010302 006104
2064 010304 006102
2065 010306 005303
2066 010310 001373
2067 010312 063700 010370
2068 010316 005504
2069 010320 063704 010372
2070 010324 005502
2071 010326 062700 001057
2072 010332 005504
2073 010334 005502
2074 010336 062704 047401
2075 010342 005502
2076 010344 062702 000006
2077 010350 062700 000002
2078 010354 005504
2079 010356 010037 010370
2080 010362 010437 010372
2081 010366 000205
2082 010370 000000
2083 010372 000000
2084
2085
2086
2087 010374
2088 010374 104402 010400
2089 010414 012737 000006 000004
2090 010422 005037 000006
2091 010426 032737 100000 177570
2092 010434 001401
2093 010436 000000
2094 010440 000137 001234

;RANDOM DATA GENERATOR SUBROUTINE
;WHEN SWITCH = 0 WE COME HERE

RAND: MOV LONUM1,R0 ;SET UP R0 WITH 5 DIGITS LOW
MOV HINUM1,R4 ;SET UP R1 WITH 5 DIGITS HIGH
MOV #7,R3 ;SET UP SHIFT COUNT
CLR R2 ;CLEAR R2
SHIFT1: ASL R0 ;SHIFT R0 LEFT AND
ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO
ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2
DEC R3 ;DECREMENT R3
BNE SHIFT1 ;CONTINUE SHIFT LOOP
ADD LONUM1,R0 ;ADDN IN NUMBER TO MAKE X 129
ADC R4 ;PROPOGATE CARRY
ADD HINUM1,R4 ;ADDN IN NUMBER TO MAKE X 129
ADC R2 ;PROPOGATE CARRY
ADD #1057,R0 ;ADDN LOW CONSTANT
ADC R4 ;PROPOGATE CARRIES
ADC R2 ;PROPOGATE AGAIN
ADD #47401,R4 ;ADDN HIGH CONSTANT
ADC R2 ;PROPOGATE CARRY
ADD #6,R2 ;ADDN HIGHEST CONSTANT
ADD #2,R0 ;REPRIME R0 WITH HIGH DIGIT
ADC R4 ;PROPOGATE CARRY
MOV R0,LONUM1 ;PUT R0 BACK IN LONUM
MOV R4,HINUM1 ;PUT R1 BACK IN HINUM
EXGEN1: RTS ;RETURN TO PROGRAM
LONUM1: 0
HINUM1: 0

;TRAP OUT ROUTINE WHEN CREATING DATA BUFFER

MEM: TYPE
4S: MOV #6,#4 ;ASCIZ <15><12>"NON-X-MEM"
CLR #6 ;RESTORE
BIT #BIT15,SWR ;TRAP
BEQ 2S ;HALT?
HALT ;NO
2S: JMP #BEGIN

C05

2095
2096
2097
2098
2099
2100 010444 012737 177770 001152
2101
2102
2103
2104
2105
2106
2107
2108
2109 010452 004737 017414
2110 010456 052737 000010 001126
2111 010464 013737 001134 001232
2112 010472 013737 001122 001154
2113 010500 005037 001200
2114 010504 013737 010240 010370
2115 010512 013737 010242 010372
2116 010520 005737 001142
2117 010524 001017
2118 010526 005037 001226
2119 010532 062737 000001 001226 25:
2120 010540 001005
2121 010542 104402 000510
2122 010546 104054
2123 010550 000137 001234
2124 010554 005737 001202 35:
2125 010560 001764
2126 010562 000426
2127 010564 022737 000042 001142 15:
2128 010572 001022
2129 010574 005737 001202 CMPLP:
2130 010600 001775
2131 010602 005737 001200
2132 010606 001007
2133 010610 004537 010262
2134 010614 013701 010370
2135 010620 010637 001200
2136 010624 020411
2137 010626 005037 001200 25:
2138 010632 013701 010372
2139 010636 000404
2140 010640 013700 001142
2141 010644 016001 000300
2142
2143
2144
2145
2146
2147
2148
2149
2150 010650 005737 001024

; THIS ROUTINE COMPARES THE DATA READ AGAINST THE DATA EXPECTED.
; ALL ERRORS ARE REPORTED TO THE OPERATOR. IF BIT 4 OF THE SWITCH
; REGISTER IS SET, THIS ROUTINE WILL CONTINUE COMPARING AFTER AN ERROR HAS BEEN
; FOUND AND WILL REPORT UP TO 8 VERIFY ERRORS WITHIN THE SAME INPUT OPERATION.
COMPAR: MOV #10, ERCOUNT ; ERROR RETRY COUNTER
; XYZ*****?
; *****
; XYZ*****?
; *****
; XYZ*****?
; *****
JSR PC, MPMSET ; PRESET MEMORY MANAGEMENT REG
BIS #BIT3, FLAG2 ; DOING COMPARE
MOV WROCT, WORK2 ; GET THE WORD COUNT
MOV VADDR, SAVE ; SET UP OUTBUFFER POINTER
CLR SWITCH ; CLEAR RANDOM PATTERN FLAG
MOV LOSAV, LONUM1 ; GET RANDOM BASE NOS.
MOV HISAV, HINUM1
TST PATNU ; TEST FOR PATTERN 0
BNE IS ; NO
CLR WORK ; CLEAR COUNTER
ADD #1, WORK ; INC COUNTER
BNE 35 ; INTERRUPT YET?
TYPE #NOINT
HLT !DA!WC!DS
JMP @BEGIN
35: TST INTFLG ; TEST FOR INT
BEQ 25 ; WAIT FOR INT BEFORE COMPARING
BR CMPLP1 ; CONT
15: CMP #42, PATNU ; IS THIS RANDOM PATTERN?
BNE CMPLP1 ; BRANCH IF YES
CMPLP: TST INTFLG ; INTERRUPT YET?
BEQ CMPLP ; NO WAIT
TST SWITCH
BNE 25
JSR RS, RAND
MOV LONUM1, GOOD ; GET EVEN RANDOM WORD
MOV SP, SWITCH ; SET RANDOM PATTERN FLAG
BR WROCMP
25: CLR SWITCH
MOV HINUM1, GOOD
BR WROCMP
CMPLP1: MOV PATNU, RD
MOV PATO(RD), GOOD
; XYZ*****?
; *****
; XYZ*****?
; *****
; XYZ*****?
; *****
WROCMP: TST TEMP2

D05

2151	010654	001406			BEQ	1\$		
2152	010656	042737	160000	001154	BIC	#160000,SAVE	;CLEAR PAR REG	
2153	010664	052737	060000	001154	BIS	#60000,SAVE	;SET PAR 3	
2154	010672	160177	170256		IS:	SUB	GOOD,2\$SAVE	;COMPARE DATA
2155	010676	0C1037			BNE	WDERR	;WORD IN ERROR	
2156	010700	005337	001232		WRDINC:	DEC	WORK2	;DECREMENT THE WORD COUNT
2157	010704	001430			BEQ	ADAM	;EXIT ROUTINE IF ZERO	
2158	010706	062737	000002	001154	ADD	#2,SAVE	;UPDATE PATTERN ADDRESS	
2159					:XYZ*****?*****			
2160					:*****			
2161					:*****			
2162					:XYZ*****?*****			
2163					:*****			
2164					:*****			
2165					:XYZ*****?*****			
2166					:*****			
2167	010714	005737	001024		TST	TEMP2		
2168	010720	001415			BEQ	1\$		
2169	010722	032737	100000	001154	BIT	#100000,SAVE	;IS 4K DONE	
2170	010730	001411			BEQ	1\$;BRANCH IF 4K NOT DONE	
2171	010732	042737	100000	001154	BIC	#100000,SAVE	;CLEAR PAR BITS	
2172	010740	052737	060000	001154	BIS	#60000,SAVE	;SET PAR 3	
2173	010746	062737	000200	172346	ADD	#200,KIPAR3	;UPDATE PAR 3 TO NEXT 4K	
2174	010754	022737	000042	001142	IS:	CMP	#42,PATNU	;IS THIS RANDOM PATTERN
2175	010762	001704			BEQ	CMPLP	;BRANCH IF YES	
2176	010764	000731			BR	WRDCMP	;COMPARE NEXT WORD	
2177	010766	042737	000010	001126	ADAM:	BIC	#BIT3,FLAG2	;DONE WITH COMPARE
2178	010774	000205			RTS	RS	;EXIT THIS ROUTINE	

E05

2179	010776	005737	001202		WDERR: TST	INTFLG	; DID INTERRUPT OCCUR YET?
2180	011002	001722			BEQ	WRDCMP	; BRANCH IF NO
2181	011004	032737	000100	177570	BIT	#BIT6, SWR	; TRY ALL?
2182	011012	001006			BNE	10\$; YES
2183	011014	005737	001204		TST	LOPCNT	; FIRST READ ERROR?
2184	011020	001403			BEQ	10\$; YES
2185	011022	005777	170004		TST	RS03	; ANY ERRORS?
2186	011026	100757			BMI	ADAM	; YES DO NOT COMPARE
2187	011030	060177	170120		10\$: ADD	GOOD, ASAVE	
2188	011034	017700	170114		MOV	ASAVE, BR)	; GET GOOD DATA
2189	011040	104436			LOGC		; LOG COMPARE ERROR
2190	011042	032737	001000	177570	BIT	#BIT9, SWR	; LOOP ON ERROR?
2191	011050	001401			BEQ	11\$; NO
2192	011052	005726			TST	(6)+	; YES UPDATE SP
2193	011054	004737	014134		11\$: JSR	PC, PRNT	; TYPEOUT?
2194	011060	001011			BNE	3\$; NO
2195	011062	104402	011066		TYPE	..+2	; .ASCIZ <15><12>"COMPARE ERR"
2196	011104	104000			3\$: HLT		; DATA COMPARE ERROR
2197	011106	004737	014134		JSR	PC, PRNT	; HAD TO DO IT THIS WAY SO
2198	011112	001022			BNE	13\$; PROGRAM COULD LOOP ON ERROR
2199	011114	104402	011120		TYPE	..+2	; .ASCIZ " ADDR="
2200	011130	005737	001216		TST	MMAVA	; IS MEM MGMT ON?
2201	011134	001406			BEQ	12\$; NO
2202	011136	013746	177776		MOV	PS, -(6)	; GET PS
2203	011142	013746	001154		MOV	SAVE, -(6)	; GET VIRTUAL ADDR
2204	011146	104412			TYPEA		; CONVERT TO PHY AND TYPE
2205	011150	000403			BR	13\$; CONT
2206	011152	013746	001154		12\$: MOV	SAVE, -(6)	; GET ADDR
2207	011156	104406			TYPES		; TYPE IT
2208	011160	005037	001160		13\$: CLR	BLOCK	; CLEAR THE BLOCK COUNTER
2209	011164	013702	001134		MOV	WRDCT, R2	; GET THE WORD COUNT
2210	011170	005202			INC	R2	; CORRECT FOR DA CALCULATIONS
2211	011172	163702	001232		MOV	WORK2, R2	; DETERMINE DISTANCE OF FAILURE INTO BUFFER
2212	011176	005737	001172		2\$: TST	RS0401	; RS04?
2213	011202	001403			BEQ	7\$; NO
2214	011204	162702	000200		SUB	#200, R2	; RS03
2215	011210	000402			BR	9\$; CONT
2216	011212	162702	000100		7\$: SUB	#100, R2	
2217	011216	100403			9\$: BMI	8\$	
2218	011220	005237	001160		INC	BLOCK	; UPDATE BLOCK COUNT FOR EACH 400 WORDS
2219	011224	000764			BR	2\$	

F05

```
2220 011226 005737 001172      8S:  TST      R50401      ;R504?
2221 011232 001403              BEQ      4S          ;NO
2222 011234 062702 000200      ADD      #200,R2    ;R504
2223 011240 000402              BR       6S          ;CONT
2224 011242 062702 000100      4S:  ADD      #100,R2    ;RESTORE POSITIVE NUMBER
2225 011246 013737 001140 001230 6S:  MOV      DMA,WORK1 ;GET HEAD AND SECTOR ADDRESS
2226 011254 063737 001160 001230 5S:  ADD      BLOCK,WORK1
2227 011262 004737 014134      JSR      PC,PRNT    ;TYPEOUT?
2228 011266 001014              BNE     1S          ;NO
2229 011270 104402 011274      TYPE    ..+2       ;.ASCIZ = DA="
2230 011302 013746 001230      MOV      WORK1,-(6) ;PUT WORK1 ON STACK
2231 011306 104406              TYPES    ;TYPE STACK IN OCTAL - SUPRESS
2232 011310 104402 011314      TYPE    ..+2       ;.ASCIZ <15><12>
2233 011320 032737 000020 177570 1S:  BIT      #BIT4,SWR   ;RETRY?
2234 011326 001405              BEQ     CLEAR      ;NO
2235 011330 005237 001152      INC     ERCOUNT    ;UPDATE ERROR COUNTER
2236 011334 001402              BEQ     CLEAR
2237 011336 000137 013700      JMP     WRDINC
2238 011342 032737 000004 001132 CLEAR: BIT      #BIT2,FLAG ;XFER TEST?
2239 011350 001404              BEQ     3S         ;NO
2240 011352 032737 010000 001132 BIT      #BIT12,FLAG ;READ?
2241 011360 001421              BEQ     2S         ;NO
2242 ;XYZ*****?*****
2243 ;*****
2244 ;*****
2245 ;XYZ*****?*****
2246 ;*****
2247 ;*****
2248 ;XYZ*****?*****
2249 ;*****
2250 011362 013701 001122      3S:  MOV      VADDR,R1  ;GET STARTING ADDR OF BUFFER
2251 011366 013700 001134      MOV      WRDCT,R0   ;NOW
2252 011372 004737 017414      JSR      PC,MMSET   ;PRESET MM
2253 011376 005737 001024      TST     TEMP2       ;IS IT TEST 2
2254 011402 001402              BEQ     1S          ;BRANCH IF NOT TEST 2
2255 011404 004537 017352      4S:  JSR      R5,MMUSE  ;DO NEXT INSTRUCTION WITH MM
2256 011410 005021      1S:  CLR     (R1)+      ;CLEAR BUFFER
2257 011412 005300      DEC     R0          ;COUNT LOCATIONS
2258 011414 001373      BNE     4S          ;WAIT TILL DONE
2259 011416 042737 000010 001126 2S:  BIC     #BIT3,FLAG2 ;DONE WITH COMPARE
2260 011424 000205      RTS     R5          ;NOW GET OUT
2261
2262 011426 013737 001076 017350 APORT: MOV      SAVAST,OUTBUF ;SET STARTING ADDR FOR OUTBUF
2263 011434 013737 001076 001122      MOV      SAVAST,VADDR ;SAVE OUTBUF ADDR
2264 011442 005737 001216      TST     MMVA        ;MEM MGMT?
2265 011446 001411              BEQ     EXTT        ;NO
2266 011450 013702 001104      MOV      SAVMGA,R2  ;SET UP MEM MGMT
2267 011454 004737 011672      MMSET: JSR      PC,STM42 ;SETUP MEM MGMT
2268 011460 010237 001122      MOV     R2,VADDR
2269 011464 013737 001124 017350      MOV     PHADDR,OUTBUF
2270 011472 000207      EXTT:  RTS     PC
```

G05

```

2271 ;TYPE CAN NOT WRITE BLOCK
2272
2273 011474 004737 014134 WTNO: JSR PC,PRNT ;TYPEOUT?
2274 011500 001002 BNE 1$ ;NO
2275 011502 104402 006724 TYPE ,NOWRIT
2276 011506 005037 001204 1$: CLR LOPCNT ;CLEAR ERR COUNTER
2277 011512 000207 RTS PC
2278
2279 ;ROUTINE TO SET UP STARTING ADDRESS FOR ALL PORTS
2280 ;AND TO CREATE WORD COUNT MAX= 20K
2281
2282 011514 013702 001074 EXTMEM: MOV STAMEM,R2 ;GET BANK #
2283 011520 005702 TST R2 ;DID HE TYPE 0?
2284 011522 001001 BNE 3$ ;NO
2285 011524 005202 INC R2 ;YES MAKE 1
2286 011526 005737 001216 3$: TST MMAVA
2287 011532 001006 BNE 1$ ;BRANCH IF MEM MGMT AVAILABLE
2288 011534 000241 CLC
2289 011536 004737 011610 JSR PC,RRR2
2290 011542 010237 001076 MOV R2,SAVAST ;SAVE A STARTING ADDR
2291 011546 000404 BR 2$ ;GET WC
2292 011550 000302 1$: SWAB R2
2293 011552 006002 ROR R2
2294 011554 010237 001104 MOV R2,SAVMGA ;SAVE ADDR FOR A PORT
2295 011560 013702 001112 2$: MOV SIZEAP,R2 ;GET 4K BLOCK COUNT
2296 011564 000241 8$: CLC
2297 011566 006002 ROR R2 ;NO CONVERT TO WC
2298 011570 004737 011610 JSR PC,RRR2
2299 011574 042702 000077 BIC #7,R2 ;CLEAR BLOCK COUNT
2300 011600 010237 001150 MOV R2,SWRDCT ;SAVE -A- PORT WC
2301 011604 000400 BR 7$ ;CONT
2302 ;4$:MOV#60000,SWRDCT;MAKE 20K
2303 011606 000207 7$: RTS PC
2304
2305 011610 006002 RRR2: ROR R2
2306 011612 006002 ROR R2
2307 011614 006002 ROR R2
2308 011616 006002 ROR R2
2309 011620 000207 RTS PC
2310
2311 011622 032737 000200 177570 WATT: BIT #BIT7,SWR ;WAIT IN BACKGROUND?
2312 011630 001003 BNE 1$ ;NO
2313 011632 004737 012176 JSR PC,XWAIT ;YES
2314 011636 000401 BR 2$ ;CONT
2315 011640 000001 1$: WAIT
2316 011642 000207 2$: RTS PC
2317
2318 011644 004737 014134 TYPREC: JSR PC,PRNT ;TYPEOUT?
2319 011650 001007 BNE 1$ ;NO
2320 011652 104402 000702 TYPE ,RECOV
2321 011656 013746 001204 MOV LOPCNT,-(6) ;GET COUNT
2322 011662 104406 TYPES ;TYPE IT
2323 011664 104402 000757 TYPE CRLFLF
2324 011670 000207 1$: RTS PC

```

H05

```
2325
2326 011672 005737 001216      STMM2: TST      MMAVA      ; MEM MGMT?
2327 011676 001002              BNE      3$          ; YES
2328 011700 000137 012164      JMP      MDON        ; GET OUT
2329 011704 005037 172340      3$: CLR      @#KIPAR0
2330 011710 010237 001154      MOV     R2,SAVE     ; SAVE R2
2331 011714 010237 172342      MOV     R2,@#KIPAR1
2332 011720 006302              ASL     R2          ; CALCULATE PHYSICAL ADDR
2333 011722 006302              ASL     R2
2334 011724 006302              ASL     R2
2335 011726 006302              ASL     R2
2336 011730 006302              ASL     R2          ; THIS BIT IS A17
2337 011732 042737 000040 001126      BIC     @BIT5,FLAG2 ; CLEAR A17?
2338 011740 103003              BCC     1$          ; SET A17
2339 011742 052737 000040 001126      BIS     @BIT5,FLAG2 ; SET BIT 5 FOR A17
2340 011750 042737 000020 001126      1$: BIC     @BIT4,FLAG2 ; CLEAR A16 FLAG
2341 011756 006302              ASL     R2          ; GET A16 BIT
2342 011760 103003              BCC     2$          ; CLEAR A16
2343 011762 052737 000020 001126      BIS     @BIT4,FLAG2 ; SET FLAG FOR A16
2344 011770 010237 001124      2$: MOV     R2,PHADDR ; GET PHYSICAL ADDR
2345 011774 013702 001154      MOV     SAVE,R2     ; SET UP MEM MGMT
2346 012000 062702 000200      ADD     @200,R2
2347 012004 010237 172344      MOV     R2,@#KIPAR2
2348 012010 062702 000200      ADD     @200,R2
2349 012014 010237 172346      MOV     R2,@#KIPAR3
2350 012020 062702 000200      ADD     @200,R2
2351 012024 010237 172350      MOV     R2,@#KIPAR4
2352 012030 062702 000200      ADD     @200,R2
2353 012034 010237 172352      MOV     R2,@#KIPAR5
2354 012040 062702 000200      ADD     @200,R2
2355 012044 010237 172354      MOV     R2,@#KIPAR6
2356 012050 012737 077406 172300      MOV     @200*256.-400+UP+RW,@#KIPDR0 ; SET KIPDR0=RW UP 200 BLOCKS
2357 012056 012737 077406 172302      MOV     @200*256.-400+UP+RW,@#KIPDR1 ; SET KIPDR1=RW UP 200 BLOCKS
2358 012064 012737 077406 172304      MOV     @200*256.-400+UP+RW,@#KIPDR2 ; SET KIPDR2=RW UP 200 BLOCKS
2359 012072 012737 077406 172306      MOV     @200*256.-400+UP+RW,@#KIPDR3 ; SET KIPDR3=RW UP 200 BLOCKS
2360 012100 012737 077406 172310      MOV     @200*256.-400+UP+RW,@#KIPDR4 ; SET KIPDR4=RW UP 200 BLOCKS
2361 012106 012737 077406 172312      MOV     @200*256.-400+UP+RW,@#KIPDR5 ; SET KIPDR5=RW UP 200 BLOCKS
2362 012114 012737 077406 172314      MOV     @200*256.-400+UP+RW,@#KIPDR6 ; SET KIPDR6=RW UP 200 BLOCKS
2363 012122 012737 077406 172316      MOV     @200*256.-400+UP+RW,@#KIPDR7 ; SET KIPDR7=RW UP 200 BLOCKS
2364 012130 012737 177600 172356      MOV     @177600,@#KIPAR7
2365 012136 012702 020000      MOV     @20000,R2
2366 012142 012737 012166 000250      MOV     @MMABTO,@MMVEC
2367 012150 012737 000020 172516      MOV     @20,SR3     ; TURN ON 22 BIT MODE
2368 012156 012737 000001 177572      MOV     @1,@#SR0   ; TURN ON MEM MGMT
2369 012164 000207      MDON: RTS      PC
2370      ; MEMORY MANAGEMENT ABORT ROUTINE FOR WRITE UP
2371 012166 104402 000541      MMABTO: TYPE     ,MTRAP
2372 012172 000000      HALT
2373 012174 000002      RTI          ; CAUSED THE ABORT
```

;BACKGROUND TEST FOR INTERRUPTS

```

2374
2375
2376 012176 052737 002000 001126 XWAIT: BIS #BIT10,FLAG2 ;WAITING IN BACKGROUND TEST
2377 012204 012737 070000 012316 MOV #70000,NPRCNT ;SETUP TIMEOUT COUNTER
2378 012212 012701 012321 MOV #NPR1+1,R1 ;SETUP WAIT LOOP
2379 012216 112711 000200 MOVB #200,(R1)
2380 012222
2381 012222 105421 2S: NEGB (R1)+
2382 012224 105441 NEGB -(R1)
2383 012226 105421 NEGB (R1)+
2384 012230 105441 NEGB -(R1)
2385 012232 105421 NEGB (R1)+
2386 012234 105441 NEGB -(R1)
2387 012236 105421 NEGB (R1)+
2388 012240 105441 NEGB -(R1)
2389 012242 105421 NEGB (R1)+
2390 012244 105441 NEGB -(R1)
2391 012246 105421 NEGB (R1)+
2392 012250 105441 NEGB -(R1)
2393 012252 105421 NEGB (R1)+
2394 012254 105441 NEGB -(R1)
2395 012256 105421 NEGB (R1)+
2396 012260 105441 NEGB -(R1)
2397 012262 102401 BVS 1S
2398 012264 000000 HALT ;ARITHMETIC OPERATION FAILED RUN DIAG
2399 012266 005337 012316 1S: DEC NPRCNT
2400 012272 001353 BNE 2S
2401 012274 104402 000510 TYPE NOINT
2402 012300 104054 HLT !DA!WC!DS
2403 012302 000137 001234 JMP @#BEGIN
2404 012306 042737 002000 001126 NPRRET: BIC #BIT10,FLAG2 ;CLEAR BKGROUND FLG
2405 012314 000207 RTS PC
2406 012316 000000 NPRCNT: 0
2407 012320 000000 NPR1: 0
2408 ;CLEAR ERROR TABLE
2409
2410 012322 012704 000020 ERRCL: MOV #20,R4 ;CLEAR
2411 012326 012703 017310 MOV #ERTAB,R3 ;ERROR
2412 012332 005023 1S: CLR (R3)+ ;TABLE
2413 012334 005304 DEC R4 ;DONE YET?
2414 012336 001375 BNE 1S ;NO
2415 012340 005037 001004 CLR PCNT ;CLEAR
2416 012344 005037 001006 CLR PCNT+2 ;PASS COUNT
2417 012350 005037 001130 CLR DROP ;CLEAR ALL DROPPED DRIVES
2418 012354 000205 RTS R5 ;RETURN

```

J05

2419
 2420
 2421
 2422
 2423
 2424
 2425
 2426
 2427
 2428
 2429
 2430
 2431
 2432
 2433
 2434
 2435
 2436
 2437
 2438
 2439
 2440
 2441
 2442
 2443
 2444
 2445
 2446
 2447
 2448
 2449
 2450
 2451
 2452
 2453
 2454
 2455
 2456
 2457
 2458
 2459

```

;RH11 POWER FAIL TEST #1
;THE STARTING ADDRESS FOR THE WRITE POWER FAIL TEST IS 270.
;A MESSAGE WILL BE TYPED OUT "LOAD SW WITH UNIT # AND CONT."
;THE OPERATOR NOW HAS TO LOAD THE UNIT # IN OCTAL INTO THE SW REGISTER
;IN BITS 00-01-AND 02. THEN HIT CONT. THE PROGRAM WILL
;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 RSER)OR A DCK
;(BIT 15 RSER). 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

012356	012706	000500		PFT1:	MOV	#500,SP		;SET UP STACK
012362	005037	001024			CLR	TEMP2		
012366	104402	000561			TYPE	,LOADSW		
012372	000000				HALT			
012374	004737	007160			JSR	PC,VECTRR		;SETUP INT VECTOR
012400	013737	177570	001164		MOV	#SWR,UNNUM		;SAVE IT
012406	004737	006770			JSR	PC,FNDTYP		;TST FOR RS03 OR 04
012412	104426			PFWATT:	CLR0V			;CLEAR ALL REG
012414	004737	013174			JSR	PC,POWFAL		;WRITE 125252 ON DISK
012420	005037	001140		PFWAT:	CLR	DMA		
012424	012737	012646	000024		MOV	#DOWN,24		;SET UP POWER FAIL VEC.
012432	012737	000340	000026		MOV	#340,26		
012440	012737	000161	001176	MYBYWR:	MOV	#161,CMD		;WRITE WITH I/E
012446	104416				DKCMD			;DO IT
012450	004737	011622			JSR	PC,WATT		;WAIT FOR INTERRUPT
012454	032737	001000	001132	3\$:	BIT	#BIT9,FLAG		;ANY ERRORS?
012462	001406				BEQ	1\$;NO
012464	104006				HLT	!DA!DB		
012466	012777	177777	166354		MOV	#-1,RSAS		;CLEAR ALL
012474	005077	166346			CLR	RSER		;ERRORS
012500	004737	007234		1\$:	JSR	PC,DISBUF		;SET UP NEW DISK BUFFER
012504	000755				BR	MYBYWR		
012506	000744				BR	PFWAT		

K05

MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 63
 DERSBB.P11 27-OCT-76 11:03 TST4 TEST FOR MULTI DISK MODE

```

2460 012510 012737 012516 001156 UPCHK: MOV #15,HRDER ;RETURN HERE IF WRONG DRIVE INTERRUPTS
2461 012516 0C5037 001140 1$: CLR DMA
2462 012522 104426 CLRQV ;INIT DRIVE
2463 012524 013737 001072 177776 CHKDAT: MOV PRIORITY,PS
2464 012532 012737 000151 001176 MOV #151,CMD ;WRITECHECK WITH I/E
2465 012540 104416 DKCMD ;DO IT
2466 012542 013737 001072 177776 MOV PRIORITY,PS
2467 012550 004737 011622 JSR PC,WATT ;WAIT FOR INTERRUPT
2468 012554 032737 001000 001132 3$: BIT #BIT9,FLAG ;ANY ERRORS?
2469 012562 0C1411 BEQ 1$ ;NO
2470 012564 104006 HLT !DB!DA
2471 012566 352737 100000 001126 BIS #BIT15,FLAG2 ;SET ERROR FLAG
2472 012574 0C5077 166246 CLR @RSER ;CLEAR ALL
2473 012600 012777 177777 166242 MOV #-1,@RSAS ;ERRORS
2474 012606 004737 007234 1$: JSR PC,DISBUF ;SET UP NEW DISK BUFFER
2475 012612 000744 BR CHKDAT
2476 012614 005737 001126 TST FLAG2 ;ANY ERRORS?
2477 012620 100405 BMI 2$ ;YES
2478 012622 104402 012626 TYPE 2$ .ASCIZ <15><12>"OK"
2479 012634 042737 100000 001126 2$: BIC #BIT15,FLAG2 ;CLEAR ERROR FLAG
2480 012642 000137 012412 JMP PFWATT ;GO WAIT FOR ANOTHER
2481 ;POWER FAIL
2482 ;POWER DOWN ROUTINE - ABORT DISK AND HALT
2483
2484
2485 012646 012737 012656 000024 DOWN: MOV #UPP,24 ;SET POWER FAIL VECTOR
2486 012654 000000 HALT
2487
2488 012656 012737 012646 000024 UPP: MOV #DOWN,24
2489 012664 012706 000500 MOV #500,5P
2490 012670 013777 001164 166136 MOV UNNUM,@RSCS2 ;GET UNIT #
2491 012676 032777 000200 166140 1$: BIT #BIT7,@RSOS ;WAIT FOR DRIVE READY
2492 012704 001774 BEQ 1$
2493 012706 000137 012510 JMP UPCHK ;GO CHECK DISK
  
```

L05

```

2494 ;POWER FAIL TEST #2
2495 ;THIS TEST WILL TEST THE SAME DRIVE THAT WAS TESTED IN THE 1ST POWER FAIL TEST
2496 ;THE PROGRAM WILL WRITE THE COMPLETE DISK WITH A 125252 PATTERN AND WILL
2497 ;THEN TELL THE OPERATOR TO POWER DOWN THE PROCESSOR.
2498 ;THE PROGRAM WILL THEN WRITE CHECK THE DISK WHILE WAITING FOR A POWER FAIL.
2499 ;WHEN THE POWER FAIL OCCURS, THE WRITE CHECKING IS ABORTED AND
2500 ;THE PPOCESSOR WILL HALT.
2501 ;THE OPERATOR SHOULD THEN TURN POWER BACK ON, THE PROGRAM WILL
2502 ;START WRITE CHECKING THE DISK AGAIN
2503 ;***NO ERRORS SHOULD OCCUR.***
2504 ;THE PROGRAM WILL TYPE OUT "OK" IF NO ERRORS OCCUR.
2505 ;THE PROGRAM WILL THEN TELL YOU TO POWER DOWN.
2506 ;DO NOT POWER OFF THE PROCESSOR AGAIN UNTIL THE PROGRAM TELLS YOU SO.
  
```

2508	012712	012706	000500			PFT2:	MOV	#500,SP	;SET UP STACK
2509	012716	005037	001024				CLR	TEMP2	
2510	012722	042737	001000	001126			BIC	#BIT9,FLAG2	;CLEAR POWER FAIL
2511	012730	012737	012752	001156			MOV	#PWRFL,HRDR	;RETURN HERE IF WRONG DRIVE INT.
2512	012736	104426					CLRQV		;INIT DRIVE
2513	012740	004737	007160				JSR	PC,VECTRR	;SETUP INT VECTOR
2514	012744	004737	013174			PWRFL2:	JSR	PC,POWFAL	;WRITE 125252 ON DISK
2515	012750	000401					BR	PWRFL	;WRITE CHECK
2516	012752	104426				PWRFL1:	CLRQV		;INIT DRIVE
2517	012754	005037	001140			PWRFL:	CLR	DMA	
2518	012760	012737	013124	000024			MOV	#PWRDN,24	;SET UP POWER FAIL VEC.
2519	012766	012737	000340	000026			MOV	#340,26	
2520	012774	013737	001072	177776		CHKDSK:	MOV	PRIORITY,PS	;ENABLE I/E
2521	013002	012737	000151	001176			MOV	#151,CMD	;WRITE CHECK WITH I/E
2522	013010	104416					DKCMD		;DO IT
2523	013012	004737	011622				JSR	PC,WATT	;WAIT FOR INTERRUPT
2524	013016	032737	001000	001132		3\$:	BIT	#BIT9,FLAG	;ANY ERRORS?
2525	013024	001411					BEQ	1\$;NO
2526	013026	104002					HLT	!DB	;YES
2527	013030	052737	100000	001126			BIS	#BIT15,FLAG2	;SET ERROR FLAG
2528	013036	005077	166004				CLR	ORSER	;CLEAR ALL
2529	013042	012777	177777	166000			MOV	#-1,ORSAS	;ERRORS
2530	013050	004737	007234			1\$:	JSR	PC,DISBUF	;CHECK NEXT BUFFER
2531	013054	000747					BR	CHKDSK	
2532	013056	032737	001000	001126			BIT	#BIT9,FLAG2	;DID POWER FAIL?
2533	013064	001733					BEQ	PWRFL	;NO
2534	013066	005737	001126				TST	FLAG2	;ANY ERRORS?
2535	013072	100405					BMI	2\$;YES
2536	013074	104402	013100				TYPE	1,+2	;ASCIZ <15><12>"OK"
2537	013106	042737	100000	001126		2\$:	BIC	#BIT15,FLAG2	;CLEAR ERRORS
2538	013114	042737	001000	001126			BIC	#BIT9,FLAG2	;CLEAR POWER FAIL FLAG
2539	013122	000710				4\$:	BR	PWRFL2	

M05

```

2540 ;ROUTINE TO ABORT DISK DURING POWER FAIL
2541
2542 013124 012737 013134 000024 PWRDN: MOV #PWRUP,24 ;SET UP RESTART
2543 013132 000000 HALT
2544
2545 013134 012737 013124 000024 PWRUP: MOV #PWRDN,24 ;RESET POWER FAIL VECTOR
2546 013142 012706 000530 MOV #500,SP
2547 013146 013777 001164 165660 MOV UNNUM,DRSCS2 ;GET UNIT #
2548 013154 052737 001000 001126 BIS #BIT9,FLAG2 ;SET POWER FAIL BIT
2549 013162 032777 000200 165654 1$: BIT #BIT7,DRSDS ;WAITING FOR
2550 013170 001774 BEQ 1$ ;DRIVE READY
2551 013172 000667 BR PWRF1 ;GO CHECK DISK
2552
2553
2554 ;ROUTINE TO WRITE THE COMPLETE DISK
2555 ;WITH 125252 PATTERN
2556 ;WRITE CHECK AND REPORT ERRORS IF THEY OCCUR
2557 ;REPORT "OK" AT COMPLETION
2558
2559 013174 012737 000020 001142 POWFAL: MOV #20,PATNU ;SET UP PATTERN
2560 013202 042737 000004 001132 BIC #BIT2,FLAG ;CLEAR XFER MODE FLAG
2561 013210 052737 010000 001126 BIS #BIT12,FLAG2
2562 013216 005037 001140 CLR DMA
2563 013222 012737 020000 017350 MOV #20000,OUTBUF ;GET STARTING ADDR FOR BUF
2564 013230 012737 020000 001122 MOV #20000,VADDR
2565 013236 012737 010000 001150 MOV #10000,SWRDCT ;SETUP WORD COUNT
2566 013244 013737 001150 001134 MOV SWRDCT,WRDCT
2567 013252 005037 001120 CLR AOB1 ;A PORT ONLY
2568 013256 013737 017350 001144 MOV OUTBUF,BUF ;SET UP CURRENT ADDRESS
2569 013264 004537 007700 JSR RS,PASEL ;GENERATE DATA BUFFER
2570 013270 012737 000161 001176 WRDNW: MOV #161,CMD ;WRITE WITH I/E
2571 013276 104416 DKCMD ;DO IT
2572 013300 004737 011622 JSR PC,WATT ;WAIT FOR INTERRUPT
2573 013304 012737 000151 001176 2$: MOV #151,CMD ;WRITECHECK I/E
2574 013312 104416 DKCMD ;DO IT
2575 013314 004737 011622 JSR PC,WATT ;WAIT FOR INTERRUPT
2576 013320 032737 001000 001132 4$: BIT #BIT9,FLAG ;ANY ERRORS?
2577 013326 001402 BEQ 1$ ;NO
2578 013330 104006 HLT !DB!DA ;YES
2579 013332 000000 HALT ;CAN NOT WRITE WITHOUT ERROR
2580 013334 004737 007234 1$: JSR PC,DISBUF ;SET UP NEW DISK BUFFER
2581 013340 000753 BR WRDNW ;WRITE NEW BUFFER
2582 013342 104402 013346 TYPE +2 ;.ASCIZ <15><12>"POWER DOWN"
2583 013364 000207 RTS PC
  
```

N05

```

2584 013366 032737 000010 177570 OUT: BIT #BIT3,SWR ;TYPEOUT ERROR COUNT?
2585 013374 001532 BEQ 15 ;NO
2586 013376 005004 CLR R4 ;CLEAR UNIT #
2587 013400 005003 CLR R3
2588 013402 053737 001130 001166 BIS DROP UNITSV ;RESTORE ALL DRIVES
2589 013410 013737 001166 001226 MOV UNITSV,WORK ;GET UNITS ON SYSTEM
2590 013416 012705 000401 MOV #401,R5 ;SETUP TEST FOR UNITS
2591 013422 030537 001226 4S: BIT R5,WORK ;IS THIS UNIT ON SYS
2592 013426 001006 BNE 2S ;YES
2593 013430 005204 5S: INC R4 ;INC UNIT #
2594 013432 010403 MOV R4,R3 ;SAVE UNIT #
2595 013434 000241 CLC
2596 013436 006105 ROL R5 ;GET NEXT DRIVE
2597 013440 103505 BCS 3S ;DONE
2598 013442 000767 BR 4S ;FIND NEXT DRIVE
2599 013444 2S:
2600 013444 104402 013450 TYPE ,+2 ;.ASCIZ <15><12>"UNIT "
2601 013460 010446 MOV R4,-(6) ;PUT R4 ON STACK
2602 013462 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2603 013464 004737 013742 JSR PC,GETERR ;GET ERROR COUNT
2604 013470 010304 MOV R3,R4 ;RESTORE UNIT #
2605 013472 104402 013476 TYPE ,+2 ;.ASCIZ <15><12>'
2606 013502 104402 000630 TYPE ,WERR ;
2607 013506 104402 013512 TYPE ,+2 ;.ASCIZ "S "
2608 013516 013746 001206 MOV WRITER,-(6) ;PUT WRITER ON STACK
2609 013522 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2610 013524 104402 013530 TYPE ,+2 ;.ASCIZ <15><12>'
2611 013534 104402 000657 TYPE ,RDERR ;
2612 013540 104402 013544 TYPE ,+2 ;.ASCIZ "S "
2613 013550 013746 001212 MOV READER,-(6) ;PUT READER ON STACK
2614 013554 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2615 013556 104402 013562 TYPE ,+2 ;.ASCIZ <15><12>'
2616 013566 104402 000642 TYPE ,WCERR ;
2617 013572 104402 013576 TYPE ,+2 ;.ASCIZ "S "
2618 013602 013746 001210 MOV WCERR,-(6) ;PUT WCERR ON STACK
2619 013606 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2620 013610 104402 013614 TYPE ,+2 ;.ASCIZ <15><12>"COMPARE ERRS "
2621 013634 013746 001214 MOV COMERR,-(6) ;PUT COMERR ON STACK
2622 013640 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2623 013642 104402 013646 TYPE ,+2 ;.ASCIZ <15><12>'
2624 013652 000666 BR 5S ;GET NEXT DRIVE
2625 013654 043737 001130 001166 3S: BIC DROP,UNITSV ;REDROP DRIVES
2626 013662 062706 000002 1S: ADD #2,SP ;RESTORE SP DUE TO JMP EXIT FROM JSR ROUTINE
2627 013666 005137 001120 COM AOB1 ;SET A OR B PORT FLAG
2628 013672 032737 000040 177570 BIT #BIT5,SWR ;TYPEOUT PASS COUNT?
2629 013700 001035 BNE DONE ;NO
2630 013702 104402 013706 TYPE ,+2 ;.ASCIZ <15><12>"END PASS "
2631 013722 013746 001006 MOV PCNT+2,-(6) ;PUT PCNT+2 ON STACK
2632 013726 104406 TYPES ;TYPE STACK IN OCTAL - SUPRESS
2633 013730 104402 013734 TYPE ,+2 ;.ASCIZ <15><12>'
2634 013740 000415 BR DONE
    
```

```

2635 013742 006304          GETERR: ASL      R4          ;GET LOC IN
2636 013744 006304          ASL      R4          ;ERR TABLE
2637 013746 062704 017310    ADD      #ERTAB,R4
2638 013752 112437 001206    MOV8    (R4)+,WRITER ;GET WRITE ERRS
2639 013756 112437 001212    MOV3    (R4)+,READER ;GET READ ERRS
2640 013762 112437 001210    MOV3    (R4)+,WCERR  ;GET WRITE CK ERRS
2641 013766 112437 001214    MOV8    (R4)+,COMERR ;GET COMPARE ERRS
2642 013772 000207    RTS
2643
2644
2645          .SBTTL          $DONE - BELL AND SCOPE ROUTINE
2646 013774 104400    DONE:   SCOPE          ;TERMINATING SCOPE FOR LOOPING
2647 013776 062737 000001 001006    ADD      #1,PCNT+2    ;ADD 1 TO THE PASS COUNT
2648 014004 005537 001004    ADC      PCNT         ;MAKE IT DOUBLE PREC.
2649 014010 013700 000042    4S:     MOV      #42,R0 ;GET MONITOR ADDRESS
2650 014014 001405    BEQ     $END1         ;IF NONE
2651 014016 000005    RESET
2652 014020 004710    SENDAO: JSR      7,(0) ;GO TO MONITOR
2653 014022 000240 000240    240,240,240 ;SAVE ROOM FOR ACT11
2654 014030 000137 003226    SEND1:  JMP      ADTST ;RETURN
2655
2656 014034 000000    .TBIT:  0            ;T BIT FLAG
2657
2658 014036 012702 000001    .LOGW:  MOV      #1,R2 ;LOG WRITE ERR
2659 014042 005003    CLIND:  CLR      R3     ;CLEAR INDEX FOR TABLE
2660 014044 000413    BR
2661
2662 014046 012702 000400    .LOGR:  MOV      #400,R2 ;LOG WRITE ERR
2663 014052 000773    BR      CLIND
2664
2665 014054 012702 000001    .LOGWC: MOV      #1,R2 ;LOG WRITE CK ERR
2666 014060 012703 000002    SETINC: MOV      #2,R3 ;SET INDEX FOR NEXT WD
2667 014064 000403    BR      ADDR
2668
2669 014066 012702 000400    .LOGC:  MOV      #400,R2 ;LOG COMPARE ERR
2670 014072 000772    BR      SETIND
2671
2672 014074 005737 001204    ADDR:   TST      LOPCNT ;1ST TIME ERROR?
2673 014100 001014    BNE
2674 014102 013704 001164    MOV      UNUM,R4     ;NO DO NOT COUNT IT
2675 014106 006304    ASL     R4          ;GET
2676 014110 006304    ASL     R4          ;POSITION IN
2677 014112 060304    ADD     R3,R4       ;ERR TABLE
2678 014114 060264 017310    ADD     R2,ERTAB(R4) ;TO ADD ERROR
2679 014120 004737 014134    JSR     PC,PRNT     ;TYPEOUT?
2680 014124 001402    BEQ     $S         ;YES
2681 014126 004737 015002    JSR     PC,DRP     ;SHOULD I DROP DRIVE?
2682 014132 000002    $S:    RTI
2683
2684 014134 032737 020000 177570    PRNT:   BIT      #BIT13,SWR ;INHIBIT TYPEOUT?
2685 014142 000207    RTS      PC
    
```

COB

MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 68
 DERSBB.P11 27-OCT-76 11:03 SCOPE - BELL AND SCOPE ROUTINE

2696	014144	052737	000004	001126	RSREG:	BIS	#BIT2,FLAG2	:SET ERROR FLAG
2697	014152	005737	016156			TST	.HLTCT	:SHOULD WE TYPE GOOD AND BAD
2698	014156	001017				BNE	BS	:NO
2699	014160	104402	014164			TYPE	.+2	:ASCIZ "BAD="
2690	014172	010046				MOV	BAD,-(6)	:PUT BAD ON STACK
2691	014174	104404				TYPEO		:TYPE STACK IN OCTAL
2692	014176	104402	014202			TYPE	.+2	:ASCIZ "GOOD="
2693	014212	010146				MOV	GOOD,-(6)	:PUT GOOD ON STACK
2694	014214	104404				TYPEO		:TYPE STACK IN OCTAL
2695	014216				BS:			
2696	014216	104402	014222			TYPE	.+2	:ASCIZ "CS1="
2697	014230	017746	164576			MOV	RSCS1,-(6)	:PUT RSCS1 ON STACK
2698	014234	104404				TYPEO		:TYPE STACK IN OCTAL
2699	014236				IS:			
2700	014236	104402	014242			TYPE	.+2	:ASCIZ "ER="
2701	014250	017746	164572			MOV	RSER,-(6)	:PUT RSER ON STACK
2702	014254	104404				TYPEO		:TYPE STACK IN OCTAL
2703	014256				2S:			
2704	014256	104402	014262			TYPE	.+2	:ASCIZ "CS2="
2705	014270	017746	164540			MOV	RSCS2,-(6)	:PUT RSCS2 ON STACK
2706	014274	104404				TYPEO		:TYPE STACK IN OCTAL
2707	014276	104402	014302			TYPE	.+2	:ASCIZ <15><12>""
2708	014306	104402	014312			TYPE	.+2	:ASCIZ "CS3="
2709	014320	017746	164540			MOV	RSCS3,-(6)	:PUT RSCS3 ON STACK
2710	014324	104404				TYPEO		:TYPE STACK IN OCTAL
2711	014326	104402	014332			TYPE	.+2	:ASCIZ "BAE="
2712	014340	017746	164516			MOV	RSBAE,-(6)	:PUT RSBAE ON STACK
2713	014344	104404				TYPEO		:TYPE STACK IN OCTAL
2714	014346	104402	014352			TYPE	.+2	:ASCIZ <15><12>""
2715	014356	032737	000200	016156		BIT	#200,.HLTCT	:PRINT SECOND SET ?
2716	014364	001112				BNE	SEEC	:YES
2717	014366	032737	000100	016156		BIT	#AS,.HLTCT	:PRINT ER ?
2718	014374	001410				BEQ	3S	:NO
2719	014376	104402	014402			TYPE	.+2	:ASCIZ "AS="
2720	014410	017746	164434			MOV	RSAS,-(6)	:PUT RSAS ON STACK
2721	014414	104404				TYPEO		:TYPE STACK IN OCTAL
2722	014416	032737	000020	016156	3S:	BIT	#BA,.HLTCT	:PRINT BUS ADDRESS
2723	014424	001410				BEQ	4S	:NO
2724	014426	104402	014432			TYPE	.+2	:ASCIZ "BA="
2725	014440	017746	164374			MOV	RSBA,-(6)	:PUT RSBA ON STACK
2726	014444	104404				TYPEO		:TYPE STACK IN OCTAL
2727	014446	032737	000004	016156	4S:	BIT	#DA,.HLTCT	:PRINT DA ?
2728	014454	001410				BEQ	5S	:NO
2729	014456	104402	014462			TYPE	.+2	:ASCIZ "DA="
2730	014470	017746	164346			MOV	RSDA,-(6)	:PUT RSDA ON STACK
2731	014474	104404				TYPEO		:TYPE STACK IN OCTAL
2732	014476	032737	000010	016156	5S:	BIT	#WC,.HLTCT	:PRINT WC ?
2733	014504	001410				BEQ	6S	:NO
2734	014506	104402	014512			TYPE	.+2	:ASCIZ "WC="
2735	014520	017746	164312			MOV	RSWC,-(6)	:PUT RSWC ON STACK
2736	014524	104404				TYPEO		:TYPE STACK IN OCTAL
2737	014526	032737	000040	016156	6S:	BIT	#DS,.HLTCT	:DRIVE STATUS
2738	014534	001410				BEQ	9S	:NO
2739	014536	104402	014542			TYPE	.+2	:ASCIZ "DS="
2740	014550	017746	164270			MOV	RSDS,-(6)	:PUT RSDS ON STACK
2741	014554	104404				TYPEO		:TYPE STACK IN OCTAL

006

MAINDEC-11-DERSB-A RM11-RSC3 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 69
DERS88.P11 27-OCT-76 11:03 SDONE - BELL AND SCOPE ROUTINE

2742 014556 032737 000002 016156 95: BIT #DB,.HLTCT ;PRINT DATA BUFFER

E06

2743	014564	001461				BEG	PTDONE		: NO
2744	014566	104402	014572			TYPE	+.2		: .ASCIZ "DB="
2745	014600	017746	164250			MOV	DRSDB, -(6)		: PUT DRSDB ON STACK
2746	014604	104404				TYPEO			: TYPE STACK IN OCTAL
2747	014606	000137	014730			JMP	PTDONE		: GET OUT
2748	014612	042737	000200	016156	SEEC:	BIC	#200, .HLTCT		: CLEAR COMMON BIT
2749	014620	032737	000240	016156		BIT	#DT, .HLTCT		: PRINT DRIVE TYPE?
2750	014626	001410				BEG	10\$: NO
2751	014630	104402	014634			TYPE	+.2		: .ASCIZ "DT="
2752	014642	017746	164212			MOV	DRSDT, -(6)		: PUT DRSDT ON STACK
2753	014646	104404				TYPEO			: TYPE STACK IN OCTAL
2754	014650	032737	000220	016156	10\$:	BIT	#MR, .HLTCT		: PRINT MR?
2755	014656	001410				BEG	11\$: NO
2756	014660	104402	014664			TYPE	+.2		: .ASCIZ "MR="
2757	014672	017746	164160			MOV	DRSMR, -(6)		: PUT DRSMR ON STACK
2758	014676	104404				TYPEO			: TYPE STACK IN OCTAL
2759	014700	032737	000204	016156	11\$:	BIT	#LA, .HLTCT		: PRINT LA?
2760	014706	001410				BEG	PTDONE		: NO
2761	014710	104402	014714			TYPE	+.2		: .ASCIZ "LA="
2762	014722	017746	164124			MOV	DRSLA, -(6)		: PUT DRSLA ON STACK
2763	014726	104404				TYPEO			: TYPE STACK IN OCTAL
2764	014730	032737	010000	001126	PTDONE:	BIT	#BIT12, FLAG2		: POWER FAIL TEST?
2765	014736	001111				BNE	RETT		: YES
2766	014740	104402	014744			TYPE	+.2		: .ASCIZ <15><12>"PASS "
2767	014754	013746	001006			MOV	PCNT+2, -(6)		: PUT PCNT+2 ON STACK
2768	014760	104406				TYPES			: TYPE STACK IN OCTAL - SUPRESS
2769	014762	032737	001000	177570		BIT	#BIT9, SWR		: LOOPING ON ERROR?
2770	014770	001404				BEG	DRP		: NO
2771	014772	104402	014776			TYPE	+.2		: .ASCIZ <15><12>
2772	015002	032737	000001	177570	DRP:	BIT	#BIT0, SWR		: DROP DRIVE?
2773	015010	001464				BEG	RETT		: NO
2774	015012	013704	001164			MOV	UNNUM, R4		: GET UNIT #
2775	015016	004737	013742			JSR	PC, GETERR		: GET ERRORS
2776	015022	063737	001206	001212		ADD	WRITER, READER		: ADD THE ERRORS
2777	015030	063737	001212	001210		ADD	READER, WCERR		
2778	015036	063737	001210	001214		ADD	WCERR, COMERR		
2779	015044	022737	000023	001214		CMP	#23, COMERR		: DROP DRIVE?
2780	015052	103043				BHIS	RETT		: NO
2781	015054	053737	001170	001130		BIS	UNCOMP, DROP		: DROP DRIVE
2782	015062	104402	015066			TYPE	+.2		: .ASCIZ <15><12>"DROPPED UNIT "
2783	015106	013746	001164			MOV	UNNUM, -(6)		: PUT UNNUM ON STACK
2784	015112	104406				TYPES			: TYPE STACK IN OCTAL - SUPRESS
2785	015114	104402	000757			TYPE	CRLF		
2786	015120	113703	001130			MOVB	DROP, R3		: GET DROPPED UNITS
2787	015124	113704	001166			MOVB	UNITSV, R4		: GET ALL DRIVES
2788	015130	020304				CMP	R3, R4		: ALL DRIVES DROPPED?
2789	015132	001003				BNE	2\$: NO
2790	015134	000000				HALT			: NO MORE DRIVES
2791	015136	000137	001234			JMP	#BEGIN		: RESTART TEST
2792	015142	032737	100000	001132	2\$:	BIT	#BIT15, FLAG		: DID OPERATOR SELECT PATTERN
2793	015150	001002				BNE	3\$: YES
2794	015152	005037	001142			CLR	PATNU		: NO CLEAR IT
2795	015156	000137	006204		3\$:	JMP	#EXTPPR		: GET NEXT DRIVE
2796	015162	000207			RETT:	RTS	PC		

F06

```

2797      ;ROUTINE TO RESTORE LOADER
2798 015164 013705 015210 RLDR:  MOV     LDRI,RS      ;GET FIRST ADDRESS OF WHERE LOADER IS
2799      ;TO BE RESTORED
2800      MOV     #17500,R4    ;ADDRESS WHERE LOADER IS STORED
2801 015170 012704 017500      MOV     #155,R2         ;WORD COUNT
2802 015174 012702 000155      MOV     (R4)+,(R5)+    ;RESTORE
2803 015200 012425      IS:    MOV     (R4)+,(R5)+
2804 015202 005302      DEC     R2
2805 015204 001375      BNE    1$
2806 015206 000000      HALT
2807 015210 017500      LDRI:  .WORD   17500    ;DONE
2808      ;FIRST ADDRESS WHERE LOADERS ARE SAVED
2809      PARCSR=172100
2810      PARVEC=114
2811 015212 012737 015304 000114 .MAM:  MOV     @PARSRV,@PARVEC
2812 015220 012737 000340 000116      MOV     #340,@PARVEC+2 ;SET PRI LEVEL TO 7
2813 015226 013746 000004      MOV     @#4,-(SP)      ;SAVE CURRENT ERROR VECTOR
2814 015232 013746 000006      MOV     @#6,-(SP)      ;SAVE PRIORITY LEVEL
2815 015236 012737 000006 000004      MOV     @#6,@#4
2816 015244 012737 000002 000006      MOV     @RTI,@#6
2817 015252 012700 172100      MOV     @PARCSR,R0     ;GET FIRST CSR ADDR
2818 015256 012702 000001      MOV     #1,R2
2819 015262 012720 000001      IS:    MOV     #1,(R0)+   ;SET ACTION ENABLE IF AVAILABLE
2820 015266 006302      ASL    R2              ;SHIFT AVAILABILITY INDICATOR
2821 015270 103374      BCC    1$
2822 015272 012637 000006      MOV     (SP)+,@#6      ;RESTORE ERROR VECTOR PRIORITY
2823 015276 012637 000004      MOV     (SP)+,@#4      ;AND INTERRUPT VECTOR
2824 015302 000207      RTS     PC
2825      ;PARITY MEMORY TRAP
2826      .PARSRV:
2827 015304 104402 015310      TYPE   .+2            ;.ASCIZ <15><12>"PARITY ERR"
2828 015326 052737 004000 001126      BIS    #BIT11,FLAG2   ;SET ERROR FLAG
2829 015334 104402 000757      TYPE   ,CRLF
2830 015340 104402 015344      TYPE   .+2            ;.ASCIZ "HIER="
2831 015354 013746 177742      MOV    #ERADD,-(SP)
2832 015360 104404      TYPE0
2833 015362 104402 015366      TYPE   .+2            ;.ASCIZ " LOER="
2834 015376 013746 177740      MOV    #ERADD,-(SP)
2835 015402 104404      TYPE0
2836 015404 104402 015410      TYPE   .+2            ;.ASCIZ " ME REG="
2837 015422 013746 177744      MOV    #MEMERR,-(SP)
2838 015426 104404      TYPE0
2839 015430 032737 000010 001126      BIT    #BIT3,FLAG2    ;WERE WE COMPARING DURING ERROR?
2840 015436 001422      BEQ    13$            ;NO
2841 015440 104402 015444      TYPE   .+2            ;.ASCIZ " ADDR="
2842 015454 005737 001216      TST   #MAVA          ;IS MEM MGMT ON?
2843 015460 001406      BEQ    12$            ;NO
2844 015462 013746 177776      MOV    PS,-(6)        ;GET PS
2845 015466 013746 001154      MOV    SAVE,-(6)      ;GET VIRTUAL ADDR
2846 015472 104412      TYPEA ;CONVERT TO PHY AND TYPE
2847 015474 000403      BR     13$            ;CONT
2848 015476 013746 001154      12$:  MOV    SAVE,-(6)    ;GET ADDR
2849 015502 104406      TYPES ;TYPE IT
2850 015504 032737 100000 177570 13$:  BIT    #BIT15,SWR     ;HALT ON ERROR?
2851 015512 001401      BEQ    13$            ;NO
2852 015514 000000      HALT                  ;YES
  
```

G06

MAINDEC-11-DERSB-A
DERSBB.P11 27-OCT-76

RH11-RSD3 DATA AND RELIABILITY TEST
11:03

MACY11 27(1006) 27-OCT-76 11:08 PAGE 72
SDONE - BELL AND SCOPE ROUTINE

2853 015516 012706 000500
2854 015522 000137 003242

IS: MOV #500,SP ;RESET STACK
JMP EXMFLG ;RESTART TEST

H06

```

2855          .SBTTL          STYPE - TTY TYPEOUT ROUTINE
2856
2857          ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
2858          ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ADR" - TYPES THE
2859          ;MESSAGE STARTING IN LOCATION "ADR:" 2) "TYPE CHAR" - TYPES
2860          ;THE ASCII "CHAR", AND 3) "PRINT (<15><12>"MESSAGE") - TYPES
2861          ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
2862          ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
2863          ;IS IN FILCHR+1.
2864
2865          .TYPE:  MOV      R4,-(6)          ;SAVE R4
2866                   MOV      R5,-(6)          ;SAVE R5
2867                   MOV      @4(6),R5        ;GET ADDRESS TO BE TYPED
2868                   BIT      #177400,R5      ;IS IT A TYPEN?
2869                   BNE      1$              ;NO
2870                   MOV      4(6),R5        ;GET ADDRESS OF CHARACTER
2871                   1$:  TSTB     (R5)        ;TERMINATOR?
2872                   BEQ      2$              ;GET OUT IF SO
2873                   CMPB    #12,(R5)        ;IS THE CHAR A LINE FEED
2874                   BNE      4$              ;NO - GET OUT
2875                   MOVB    FILCHR+1,R4     ;GET THE FILL COUNT
2876                   5$:  MOVB    FILCHR,@TPB  ;TYPE A FILLER
2877                   TSTB    @TPS           ;DONE YET?
2878                   BPL     .-4            ;NO - WAIT
2879                   DEC     R4              ;DEC COUNT
2880                   BNE     5$             ;LOOP UNTIL 0
2881                   4$:  MOVB    (R5)+,@TPB  ;LOAD AND TYPE THE CHARACTER
2882                   TSTB    @TPS           ;IS THE PRINTER READY
2883                   BPL     .-4            ;WAIT UNTIL IT IS
2884                   BR      1$             ;GET THE NEXT CHARACTER
2885                   2$:  MOV      @4(6),-(6)  ;GET ADDRESS TO BE TYPED
2886                   ADD     #2,6(6)        ;ADD 2 TO THE ADDRESS
2887                   CMP     (6)+,4(6)      ;IS IT .+2?
2888                   BNE     3$             ;NO
2889                   ADD     #2,R5          ;ADD 2 TO THE ADDRESS
2890                   BIC     #1,R5          ;BACK UP TO AN EVEN BYTE
2891                   MOV     R5,4(6)        ;RESTORE ADDRESS
2892                   3$:  MOV     (6)+,R5     ;RESTORE R5
2893                   MOV     (6)+,R4        ;RESTORE R4
2894                   RTI
  
```

```

2895          .SBTTL          SSCOPE - SCOPE LOOP HANDLER
2896
2897          ; THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
2898          ; LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
2899          ; "SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
2900          ; RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
2901
2902 015664 032737 040000 177570 .SCOPE: BIT      #SW14, @#SWR      ; LOOP ON TEST?
2903 015672 001045          BNE      .KIT          ; LOOP ON TEST IS SET
2904 015674 000416          BR       3$          ; SKIP - NOP FOR XOR TESTER
2905 015676 013746 000004          MOV     @#4, -(6)      ; PUSH @#4 ON STACK
2906 015702 012737 015722 000004          MOV     #4$,@#4      ; SET FOR TIMEOUT
2907 015710 005737 177060          TST    @#177060     ; ERROR ON XOR?
2908 015714 012637 000004          MOV     (6)+, @#4      ; POP STACK INTO @#4
2909 015720 000422          BR       .SVLAD      ; NO ERROR - GO TO NEXT TEST
2910 015722 022626          4$:    CMP     (6)+, (6)+      ; CLEAR STACK
2911 015724 012637 000004          MOV     (6)+, @#4      ; POP STACK INTO @#4
2912 015730 000426          BR       .KIT          ; ERROR - LOOP ON TEST
2913 015732 032737 004000 177570 3$:    BIT      #SW11, @#SWR     ; KILL ITERATIONS
2914 015740 001012          BNE      .SVLAD      ; YES - KILL ITERATIONS
2915 015742 105737 001001          TSTB   ICNT+1        ; FIRST ONE?
2916 015746 001404          BEQ     2$          ; BRANCH IF FIRST
2917 015750 123737 016034 001001          CMPB   TIMES, ICNT+1 ; DONE?
2918 015756 003013          BGT     .KIT          ; BRANCH IF NOT
2919 015760 112737 000001 001001 2$:    MOVB   #1, ICNT+1     ; FIRST ITERATION
2920 015766 105237 001000          .SVLAD: INCB   ICNT      ; COUNT TEST NUMBERS
2921 015772 011637 001010          MOV     (6), LAD      ; SAVE LOOP ADDRESS
2922 015776 013737 001000 177570          MOV     ICNT, @#DISPLAY ; DISPLAY TEST NO. AND ITERATION COUNT
2923 016004 000002          RTI                    ; RETURN
2924
2925 016006 105237 001001          .KIT:   INCB   ICNT+1      ; INC THE ITERATION COUNT
2926 016012 013737 001000 177570 .OVER:  MOV     ICNT, @#DISPLAY ; SET UP DISPLAY
2927 016020 005737 001010          TST    LAD          ; FIRST ONE?
2928 016024 001760          BEQ     .SVLAD      ; YES
2929 016026 013716 001010          MOV     LAD, (6)     ; FUDGE RETURN ADDRESS
2930 016032 000002          RTI                    ; FIXES PS
2931
2932 016034 000001          TIMES: 1              ; RUN 1 TIMES

```

JOB

```

2933          .SBTTL          $HLT - HLT ROUTINE (ERROR TYPEOUT)
2934
2935          ; THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
2936          ; ADDRESS OF THE "HLT". IT ALSO COUNTS THE NUMBER OF ERRORS
2937          ; AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
2938          ; "HALT" ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
2939          ; (HLT+3) WILL BE PLACED IN ".HLTCT:" FOR ADDITIONAL TYPEOUTS.
2940
2941 016036 005237 001002          .HLT:  INC      ERRORS      ; INC THE ERROR COUNT
2942 016042 032737 020000 177570  BIT      #SW13,2#SWR  ; SKIP TYPEOUT IF SET
2943 016050 001025                BNE      2$          ; SKIP TYPEOUTS
2944 016052 104402 016056          TYPE     .+2          ; ASCIZ (15)(12)
2945 016062 011637 001012          MOV      (6),HLTADR ; PUT ADDRESS OF INSTRUCTION ON STACK
2946 016066 162737 003002 001012  SUB      #2,HLTADR  ; FUDGE ADDRESS
2947 016074 117737 162712 016156  MOVB    2#HLTADR,.HLTCT ; GET HLT ARGUMENT
2948 016102 013746 001012          MOV      HLTADR,-(6) ; PUT HLTADR ON STACK
2949 016106 104404                TYPEO    ; TYPE STACK IN OCTAL
2950 016110 104402 016114          TYPE     .+2          ; ASCIZ " "
2951 016120 004737 014144          JSR      PC,RSREG   ; GO TO USER ERROR ROUTINE
2952 016124 005737 177570          2$:    TST      2#SWR  ; HALT ON ERROR
2953 016130 100001                BPL      .+4          ; SKIP IF CONTINUE
2954 016132 000000                HALT    ; HALT ON ERROR!
2955 016134 032737 001000 177570  BIT      #SW9,2#SWR  ; CHECK FOR INHIBIT LOOP ON ERROR
2956 016142 001003                BNE      3$          ; SKIP IF LOOP ON ERROR
2957 016144 105037 001001          CLRB    ICNT+1     ; CLEAR ITERATION COUNT
2958 016150 000002                RTI     ; RETURN
2959 016152 000137 016006          3$:    JMP      .KIT  ; LOOP ON TEST UNTIL NO ERRORS
2960
2961 016156 000000          .HLTCT: 0          ; HLT ARGUMENT
  
```

K06

MAINDEC-11-DERSB-A
DERSBB.P11

27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST
SOCIAL - OCTAL TYPEOUT ROUTINE

MACY11 27(1006) 27-OCT-76 11:08 PAGE 76

```

2962          .SBTTL          SOCIAL - OCTAL TYPEOUT ROUTINE
2963
2964          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
2965          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
2966          ;16 BITS. IT IS CALLED VIA THE TYOCT, TYBIT, OR TYOCS MACRO'S.
2967
2968 016160 012737 170101 016346 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
2969 016166 000411                BR          .PTIT          ;NOW TYPE IT IN BIT FORM
2970 016170 112737 000001 016346 .TYPEO: MOVVB   #1,.PR          ;SET ZERO FILL SWITCH
2971 016176 000402                BR          .+6          ;SKIP
2972 016200 005037 016346          .TYPES: CLR      .PR          ;SUPPRESS LEADING ZERO'S
2973 016204 112737 177772 016347          MOVVB   #-6,.PR+1      ;SET COUNT
2974 016212
2975 016212 010446                MOV      R4,-(6)      ;PUSH R4 ON STACK
2976 016214 010546                MOV      R5,-(6)      ;PUSH R5 ON STACK
2977 016216 016605 000010          MOV      10(6),R5     ;GET THE DATA
2978 016222 012704 016350          MOV      #.PR+2,R4    ;SET POINTER TO FIRST ASCII CHAR.
2979 016226 105014                CLRB     (4)          ;CLEAR FIRST BYTE
2980 016230 000411                BR          .PRF       ;ROTATE FIRST BIT
2981 016232 105014          .PRL:  CLR      (4)          ;CLEAR BYTE OF CHARACTER
2982 016234 032737 000100 016346          BIT      #100,.PR     ;BIT TYPING MODE?
2983 016242 001004                BNE      .PRF         ;YES - SKIP 2 ROTATES
2984 016244 006105                ROL      R5           ;ROTATE BIT INTO C
2985 016246 106114                ROLB     (4)          ;PACK IT
2986 016250 006105                ROL      R5           ;ROTATE BIT INTO C
2987 016252 106114                ROLB     (4)          ;PACK IT
2988 016254 006105          .PRF:  ROL      R5           ;ROTATE BIT INTO C
2989 016256 106114                ROLB     (4)          ;PACK IT
2990 016260 105714                TSTB     (4)          ;IS IT ZERO?
2991 016262 001402                BEQ      .+6          ;SKIP INC
2992 016264 105237 016346          INCB     .PR          ;SET FILL SWITCH
2993 016270 105737 016346          TSTB     .PR          ;CHECK FILL SWITCH
2994 016274 001402                BEQ      .+6          ;SKIP BITSET
2995 016276 152724 000060          BISB     #'0,(4)+     ;MAKE INTO ASCII CHAR
2996 016302 105237 016347          INCB     .PR+1        ;INC COUNT
2997 016306 001351                BNE      .PRL         ;REPEAT
2998 016310 022704 016350          CMP      #.PR+2,R4    ;EMPTY BUFFER?
2999 016314 001002                BNE      .+6          ;SKIP IF NOT
3000 016316 112724 000060          MOVVB   #'0,(4)+     ;LOAD 1 ZERO
3001 016322 105014                CLRB     (4)          ;NULL TERMINATOR
3002 016324 104402 016350          TYPE     .PR+2        ;TYPE IT
3003 016330 012605                MOV      (6)+,R5      ;POP STACK INTO R5
3004 016332 012604                MOV      (6)+,R4      ;POP STACK INTO R4
3005 016334 016666 000002 000004          MOV      2(6),4(6)    ;GET RID OF
3006 016342 012616                MOV      (6)+,(6)     ;DATA WORD
3007 016344 000002                RTI
3008
3009 016346 000012          .PR:  .BLKW   12      ;COUNT, SWITCH, AND OUTPUT BUFFER

```

L06

```
3010 .SBTTL $POWER - POWER DOWN AND UP ROUTINES
3011
3012 ; THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
3013 ; THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
3014 ; WAIT FOR POWER TO GO DOWN AND BE RESTORED.
3015 ; IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
3016 ; THE PROGRAM WILL HALT AT '.ILLUP'.
3017
3018 016372 012777 016520 000126 .POWER: MOV $ILLUP,2.PUVEC ;SET FOR FAST UP
3019 016400 012777 000340 000122 MOV $340,2.PUVECS+2 ;PRIO:7
3020 016406 010046 MOV R0,-(6) ;PUSH R0 ON STACK
3021 016410 010146 MOV R1,-(6) ;PUSH R1 ON STACK
3022 016412 010246 MOV R2,-(6) ;PUSH R2 ON STACK
3023 016414 010346 MOV R3,-(6) ;PUSH R3 ON STACK
3024 016416 010446 MOV R4,-(6) ;PUSH R4 ON STACK
3025 016420 010546 MOV R5,-(6) ;PUSH R5 ON STACK
3026 016422 010637 016524 MOV SP,SAVR6 ;SAVE SP
3027 016426 012777 016436 000072 MOV $POWUP,2.PUVEC ;SET UP VECTOR
3028 016434 000000 HALT ;WAIT FOR PF
3029
3030 016436 013706 016524 .PCWUP: MOV .SAVR6,SP ;GET SP
3031 016442 005001 CLR R1 ;WAIT LOOP FOR THE TTY
3032 016444 005201 1$: INC R1 ;WAIT FOR THE INC
3033 016446 001376 BNE 1$ ;OF WORD
3034 016450 012605 MOV (6)+,R5 ;POP STACK INTO R5
3035 016452 012604 MOV (6)+,R4 ;POP STACK INTO R4
3036 016454 012603 MOV (6)+,R3 ;POP STACK INTO R3
3037 016456 012602 MOV (6)+,R2 ;POP STACK INTO R2
3038 016460 012601 MOV (6)+,R1 ;POP STACK INTO R1
3039 016462 012600 MOV (6)+,R0 ;POP STACK INTO R0
3040 016464 012737 016372 000024 MOV $POWER,2$24 ;SET UP THE POWER DOWN VECTOR
3041 016472 012737 000340 000026 MOV $340,2$26 ;PRIO:7
3042 016500 104402 016504 TYPE +2 ;.ASCIZ <15><12>"POWER"
3043 016514 000137 007176 JMP TIMUP ;JMP TO USER ADDRESS
3044
3045 016520 000000 .ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
3046 016522 000776 BR .-2 ; BEFORE THE POWER DOWN WAS COMPLETE
3047
3048 016524 000000 .SAVR6: 0 ;PUT THE SP HERE
3049 016526 000024 000026 .PUVEC: 24,26 ;POWER UP VECTOR
```

MO6

```

3050
3051
3052
3053
3054
3055
3056
3057 016532
3058 016532 010446
3059 016534 010546
3060 016536 016605 000012
3061 016542 016604 000010
3062 016546 016666 000006 000010
3063 016554 016666 000004 000006
3064 016562 016666 000002 000004
3065 016570 012616
3066 016572 010346
3067 016574 000305
3068 016576 006005
3069 016600 006005
3070 016602 006005
3071 016604 042705 177771
3072 016610 016505 016764
3073 016614 010403
3074 016616 042704 160000
3075 016622 000303
3076 016624 006003
3077 016626 006003
3078 016630 006003
3079 016632 006003
3080 016634 042703 177761
3081 016640 060305
3082 016642 011505
3083 016644 005003
3084 016646 006305
3085 016650 006103
3086 016652 006305
3087 016654 006103
3088 016656 006305
3089 016660 006103
3090 016662 006305
3091 016664 006103
3092 016666 006305
3093 016670 006103
3094 016672 006305
3095 016674 006103
3096 016676 060405
3097 016700 005503
3098 016702 006305
3099 016704 006103
3100 016706 010346
3101 016710 000241
3102 016712 006016
3103 016714 000241
3104 016716 006016
3105 016720 000241
  
```

```

.SBTTL                    $TYPEA - 18 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 18 BIT FORM. THIS ROUTINE IS LINKED
;VIA THE 'TYPADR' MACRO.

.TYPEA:
  MOV     R4, -(6)                    ;PUSH R4 ON STACK
  MOV     R5, -(6)                    ;PUSH R5 ON STACK
  MOV     12(6), R5                   ;R5 - C D PS WITH PREVIOUS MODE
  MOV     10(6), R4                   ;R4 - ADDRESS TO BE DECODED AND TYPED
  MOV     6(6), 10(6)                ;MOVE
  MOV     4(6), 6(6)                 ;DOWN
  MOV     2(6), 4(6)                 ;FOUR
  MOV     (6)+, (6)                  ;WORDS
  MOV     R3, -(6)                    ;PUSH R3 ON STACK
  SWAB    R5                          ;GET THE
  ROR     R5                          ;2 PREVIOUS
  ROR     R5                          ;MODE BITS
  ROR     R5                          ;INTO POSITION
  BIC     #177771, R5                 ;TO USE AS AN OFFSET
  MOV     .SATAB(5), R5               ;R5 - SPACE ADDRESS FOR MM
  MOV     R4, R3                      ;R3 - REGISTER OFFSET
  BIC     #160000, R4                 ;CLEAR THE MM REG SELECT BITS
  SWAB    R3                          ;NOW MAKE
  ROR     R3                          ;MM REG
  ROR     R3                          ;SELECT BITS
  ROR     R3                          ;INTO AN
  ROR     R3                          ;OFFSET
  BIC     #177761, R3                 ;CLEAR THE JUNK BITS
  ADD     R3, R5                      ;ADD THE OFFSET TO THE TABLE
  MOV     (5), R5                    ;GET THE ISAR DATA

  ;THIS IS
  ;TO SHIFT
  ;THE SEGMENT
  ;ADDRESS
  ;AN 18 BIT
  ;ADDRESS
  ;POSITION
  ;WITH R3 CONTAINING
  ;THE UPPER 2 BITS
  ;AND R5 CONTAINING
  ;THE 16 BIT ADDRESS
  ;THEN SHIFT FOR TYPING

  MOV     R3, -(SP)
  CLC
  ROR     (SP)
  CLC
  ROR     (SP)
  CLC
  
```

N06

MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 79
DERSB8.P11 27-OCT-76 11:03 \$TYPEA - 18 BIT ADDRESS TYPED

3106	016722	005016		ROR	(SP)	
3107	016724	104406		TYPES		
3108	016726	042703	177770	BIC	#177770,R3	
3109	016732	110337	016350	MOVB	R3,.PR+2	;GET THE FIRST NUMBER FROM R3
3110	016736	062737	000060	ADD	#0,.PR+2	;MAKE IT INTO A NUMBER
3111	016744	012704	016351	MOV	#.PR+3,R4	;FUDGE IN THE POINTER
3112	016750	012737	175401	MOV	#175401,.PR	;AND THE FLAGS (FILL 8 5 BYTES)
3113	016756	012603		MOV	(6)+,R3	;POP STACK INTO R3
3114	016760	000137	016232	JMP	.PRL	;DECODE AND TYPE THE REST
3115						
3116	016764	172340		.SATAB:	172340	;KISARD

```

3117          .SBTTL          STRAP - TRAP HANDLER
3118
3119          ; THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APPROPRIATE
3120          ; SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
3121          ; THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
3122          ; FOLLOW THIS MACRO.
3123
3124          016766 011646          .TRAP:  MOV      (6),-(6)          ; GET ADDRESS OF TRAP +2
3125          016770 162716 0.0002  SUB      #2,(6)          ; MAKE IT ADDRESS OF TRAP
3126          016774 017616 003000  MOV      @ (6), (6)        ; GET TRAP INSTRUCTION
3127          017000 062716 112406  ADD      @,TRAP+2-TRAP,(6) ; GET DATA AND MAKE IT AN OFFSET
3128          017004 013607          .TRP:  MOV      @ (6)+,PC          ; GO TO PROPER SUBROUTINE
3129
3130          017006 015664          .SCOPE          ; SCOPE = TRAP+0          (104400)
3131          017010 015526          .TYPE          ; TYPE = TRAP+2          (104402)
3132          017012 016170          .TYPE0         ; TYPE0 = TRAP+4          (104404)
3133          017014 016200          .TYPES         ; TYPES = TRAP+6          (104406)
3134          017016 020556          .TYPED        ; TYPED = TRAP+10         (104410)
3135          017020 016532          .TYPEA        ; TYPEA = TRAP+12         (104412)
3136          017022 076242          .ERCLR        ; ERCLR = TRAP+14         (104414)
3137          017024 006264          .DKCMD        ; DKCMD = TRAP+16         (104416)
3138          017026 017046          .RDOCT        ; RDOCT = TRAP+20         (104420)
3139          017030 017164          .RDLIN        ; RDLIN = TRAP+22         (104422)
3140          017032 020524          .UPDAT        ; UPDAT = TRAP+24         (104424)
3141          017034 000342          .CLRDV        ; CLRDV = TRAP+26         (104426)
3142          017036 014036          .LOGW         ; LOGW = TRAP+30         (104430)
3143          017040 014046          .LOGR         ; LOGR = TRAP+32         (104432)
3144          017042 014054          .LOGWC        ; LOGWC = TRAP+34         (104434)
3145          017044 014066          .LOGC         ; LOGC = TRAP+36         (104436)
    
```


C07

```

3146          .SBTTL          SRDOCT - OCTAL INPUT ROUTINE
3147
3148          ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
3149          ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
3150
3151 017046 011646          .RDOCT: MOV      (6),-(6)          ;MOVE THE PC
3152 017050 016666 000004 000002 MOV      4(6),2(6)          ;MOVE THE PS
3153 017056 010146          MOV      R1,-(6)          ;PUSH R1 ON STACK
3154 017060 010246          MOV      R2,-(6)          ;PUSH R2 ON STACK
3155 017062 010346          MOV      R3,-(6)          ;PUSH R3 ON STACK
3156 017064 104422          4$: RDLIN          ;READ A LINE INTO INPUT
3157 017066 005001          CLR      R1          ;INIT DATA WORD
3158 017070 012703 017270 MOV      @INPUT,R3          ;INIT POINTER
3159 017074 112302          1$: MOVB   (3)+,R2          ;GET A BYTE
3160 017076 001417          BEQ     2$          ;GET OUT IF ZERO
3161 017100 122702 000060 CMPB   #'0',R2          ;CHECK FOR 0 OR GREATER
3162 017104 003022          BGT     3$          ;ERROR - LESS THAN 0
3163 017106 122702 000067 CMPB   #'7',R2          ;CHECK FOR 7 OR LESS
3164 017112 002417          BLT     3$          ;ERROR - GREATER THAN 7
3165 017114 006002          ROR     R2          ;GET
3166 017116 006002          ROR     R2          ;INTO
3167 017120 006002          ROR     R2          ;POSITION
3168 017122 006101          ROL     R1          ;FIRST BIT
3169 017124 006102          ROL     R2          ;GET
3170 017126 006101          ROL     R1          ;SECOND BIT
3171 017130 006102          ROL     R2          ;GET
3172 017132 006101          ROL     R1          ;THIRD BIT
3173 017134 000757          BR      1$          ;LOOP
3174 017136 010166 000012 2$: MOV      R1,12(6)          ;SAVE THE RESULT
3175 017142 012603          MOV      (6)+,R3          ;POP STACK INTO R3
3176 017144 012602          MOV      (6)+,R2          ;POP STACK INTO R2
3177 017146 012601          MOV      (6)+,R1          ;POP STACK INTO R1
3178 017150 000002          RTI
3179
3180          3$:
3181 017152 104402 017156          TYPE   4,+2          ;ASCIZ "'<15><12>'
3182 017162 000740          BR      4$          ;TRY AGAIN
  
```

```

3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
017164 010546
017166 012705 017270
017172 022705 017310
017176 001412
017200 105737 177560
017204 100375
017206 113715 177562
017212 142715 000200
017216 122715 000177
017222 001005
017224
017224 104402 017230
017234 000754
017236 111527 000000
017242 104402 017240
017246 122725 000015
017252 001347
017254 105065 177777
017260 104402 000012
017264 012605
017266 000002
017270 000020
017310 000020
017350 000000

```

.SBTTL SRDLIN - TTY INPUT ROUTINE

```

;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NUL TERMINATOR
;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYP YG
;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.

```

```

.RDLIN: MOV     R5, -(6)           ;SAVE R5
1$:     MOV     @INPUT, R5       ;GET ADDRESS
2$:     CMP     @INPUT+16., R5   ;BUFFER FULL?
        BEQ     4$              ;YES - TYPE "?"
        TSTB   @177560         ;WAIT FOR
        BPL     -4              ;A CHARACTER
        MOVB   @177562, (5)     ;GET CHARACTER
        BICB   @200, (5)        ;GET RID OF JUNK
        CMPB   @177, (5)        ;IS IT A RUBOUT
        BNE    3$              ;SKIP IF NOT
4$:     TYPE   ,+2              ;.ASCIZ "?"(15)<(12)
        BR     1$              ;ZAP THE BUFFER AND LOOP
3$:     MOVB   (5), #0          ;SET UP FOR TYPING
        TYPE   ,3$+2           ;ECHO IT
        CMPB   @15, (5)+       ;CHECK FOR RETURN
        BNE    2$              ;LOOP IF NOT RETURN
        CLRB   -1(5)           ;ZAP RETURN (THE 15)
        TYPE   ,12             ;TYPE A LINE FEED
        MOV    (6)+, R5        ;RESTORE R5
        RTI                    ;RETURN

```

```

INPUT:  .BLKB 16.
ERTAB:  .BLKW 16.
OUTBUF: 0
;TTY INPUT AREA

```

E07

3216
 3217
 3218
 3219
 3220
 3221
 3222
 3223
 3224
 3225
 3226
 3227
 3228
 3229
 3230
 3231
 3232
 3233
 3234
 3235
 3236
 3237
 3238
 3239
 3240
 3241
 3242
 3243
 3244
 3245
 3246
 3247
 3248
 3249
 3250
 3251
 3252
 3253
 3254
 3255
 3256
 3257
 3258
 3259
 3260
 3261
 3262
 3263
 3264
 3265

```

; THIS SUBROUTINE HAS THE CALL
; JSR PC,MMUSE
; XXX
; WHERE XXX IS AN EXECUTABLE ONE WORD INSTRUCTION
; MEM MANAGEMENT IS USED ON THE DESTINATION
; THE DESTINATION FIELD MUST BE (R1)+
; NO OTHER FORM IS ALLOWED

MMUSE:  TST     TEMP2
        BEQ     2$
        MOV     (R5)+,1$
        BIC     #160000,R1
        BIS     #60000,R1
1$:     .WORD   0 ;CONTAINS INSTRUCTION JUST AFTER
        ;THE JSR R5, MMUSE
        BIT     #100000,R1
        BEQ     2$
        ADD     #200,KIPAR3
2$:     RTS    R5
  
```

```

; THIS SETUP MEM MANAGEMENT FOR A FOLLOWING INSTRUCTION
; CALL JSR PC,MMPSET
  
```

```

MMPSET: TST     TEMP2
        BEQ     2$
        MOV     TEMP3,-(6)
        CLR     KIPAR3
1$:     ADD     #200,KIPAR3
        DEC     (6)
        BNE     1$
        TST     (6)+
2$:     RTS    PC

TIEOUT: TYPE    ,CRLF
        TYPE    ,+2 ;.ASCIZ "TIMEOUT PC="
        SUB     #2,-(SP)
        TYPE0
        HALT
  
```

;NOTE FOR PROGRAMMER***** PROGRAM AT THIS POINT CAN NOT EXCEED A PC OF 17500*****

F07

MAINDEC-11-DERSB-A
DERSB8.P11

27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST
SRDLIN - TTY INPUT ROUTINE

MACY11 27(1006) 27-OCT-76 11:08 PAGE 84

```
3266          020000          = 20000
3267          ;NOTE ALL THIS CODE GETS DESTROYED WHEN PATTERN IS WRITTEN
3268
3269          ;ROUTINE TO SAVE ABS LOADER
3270 020000 012700 017776      LDR:  MOV #17776,R0
3271 020004 012737 020024 000004  MOV #25,4 ;SET TIME OUT TRAP VECTOR
3272 020012 012737 000340 000006  MOV #340,6
3273 020020 005720          TST (R0)+
3274 020022 000776          BR -2
3275 020024 022626          25:  CMP (SP)+,(SP)+
3276 020026 012737 000006 000004  MOV #6,4
3277 020034 005037 000006          CLR 6
3278 020040 162700 000334          SUB #334,R0 ;POINT R0 BACK TO LOADER
3279 020044 010037 015210          MOV R0,LDR1 ;SAVE FOR RESTORE ROUTINE
3280 020050 012702 000155          MOV #155,R2 ;WORD COUNT
3281 020054 012703 017500          MOV #17500,R3 ;WHERE LOADER IS TO BE STORED
3282 020060 012023          15:  MOV (R0)+,(R3)+ ;STORE LOADER
3283 020062 005302          DEC R2
3284 020064 001375          BNE 15
3285 020066 000207          RTS PC ;RETURN
3286
3287
3288          ; -A- PORT SIZE
3289
3290 020070 052737 020000 001126  SIZZAP: BIS #BIT13,FLAG2 ;SET MAPPING BIT
3291 020076 004737 001620          JSR PC,DRVENO ;FIND DRIVE
3292 020102 012737 000002 001230  MOV #2,WORK1 ;START WITH ONE 4K BUFFER
3293 020110 012737 000001 001074  MOV #1,STAMEM ;FIRST 4K BANK
3294 020116 012737 057476 001144  MOV #57476,BUF ;GET STARTING ADDR. 5K
3295 020124 012737 000001 001134  MOV #1,WRDCT ;LOAD WC
3296 020132 005037 001140          CLR DMA ;LOAD DA
3297 020136 012777 000040 160670  MOV #40,DRSCS2 ;CLEAR ALL RS REG
3298 020144 013777 001164 160662  MOV UNNUM,DRSCS2 ;GET DRIVE #
3299
3300          :XYZ*****?*****
3301          :*****
3302          :XYZ*****?*****
3303          :*****
3304          :XYZ*****?*****
3305          :*****
3306
3307 020152 005037 001026          CLR TEMP3
3308 020156 012737 000061 001176  45:  MOV #61,CMD ;DO A ERITE
3309 020164 004416          15:  DKCMD ;NOW
3310 020166 005777 160640          TSTB DRSCS1 ;DONE YET?
3311 020172 000375          BPL 15 ;NO
3312 020174 032777 004000 160632  BIT #4000,DRSCS2 ;DID NEM SET?
3313 020202 001012          BNE SIZ1 ;YES
3314 020204 005777 160622          TST DRSCS1 ;ANY ERRORS?
3315 020210 100005          BPL 35 ;NO
3316 020212 012737 000006 001112  MOV #6,SIZZAP ;GET SIZE OF BUFFER
3317 020220 000137 020326          JMP #SIZERR ;FOR USER IF HE WISHES IT
3318 020224 104424          35:  UPDAT ;GET NEXT 4K BANK
3319 020226 000756          BR 45 ;TEST BANK
3320 020230 005337 001230          SIZ1: DEC WORK1 ;DEC SIZE OF BUFFER
3321 020234 013737 001230 001112  MOV WORK1,SIZZAP ;LOAD SIZE OF A BUFFER
```

G07

MAINDEC-11-DERSB-A
DERSBB.P11

27-OCT-76 11:03

RH11-RS03 DATA AND RELIABILITY TEST
SRDLIN - TTY INPUT ROUTINE

MACY11 27(1006) 27-OCT-76 11:08 PAGE 85

3322				
3323				
3324				
3325				
3326				
3327				
3328				
3329				
3330	020242	013737	001230	001022
3331	020250	104402	020254	
3332	020316	004737	020464	

```

;XYZ*****?*****
;*****
;*****
;XYZ*****?*****
;*****
;*****
;XYZ*****?*****
;*****

```

```

MOV      WORK1,TEMP1
TYPE    PC,+2      ;.ASCIZ <15><12> "PORT -A- DATA BUFFER 4K TO "
JSR     PC,SIZPR

```

H07

```
3333 020322 000137 020444          SIZZBP: JMP      NOB          ;GET OUT NO -B- PORT
3334 020326                                SIZERR:
3335 020326 104402 020332          TYPE      .+2          ;.ASCIZ <15><12>"WILL NOT CONTINUE TO SIZE MEMORY BECAUS
3336 020416 012737 000006 001114    MOV      #6,SIZEBP      ;GIVE PROGRAM A BUFFER
3337 020424 104060                                HLT      !BADDS        ;YOU CAN ENTER CONVERSATION MODE
3338 020426 052737 000001 001126    BIS      #BIT0,FLAG2    ;BEEN HERE BEFORE FLAG
3339 020434 042737 020000 001126    BIC      #BIT13,FLAG2   ;CLEAR MAPPING FLAG
3340 020442 000000                                HALT
3341 020444 042737 020000 001126    NOB:     BIC      #BIT13,FLAG2 ;CLEAR MAPPING FLAG
3342 020452 052737 000002 001126    BIS      #BIT1,FLAG2    ;SET BEEN HERE FLAG
3343 020460 000137 001462          JMP      CALM          ;CAL BUFFER AND WC
3344
3345 020464 005001          SIZPR: CLR      R1          ;INIT SETUP
3346 020466 012702 000004          MOV      #4,R2
3347 020472 062701 000001          SIZP:   ADD     #1,R1      ;SETUP FOR BANK NO
3348 020476 062702 000004          ADD     #4,R2          ;SETUP FOR SIZE FO MEMORY
3349 020502 020137 001230          CMP     R1,WORK1      ;IS THIS THE RIGHT SIZE?
3350 020506 001371          BNE     SIZP          ;NO
3351 020510 010246          MOV     R2,-(6)      ;PUT R2 ON STACK
3352 020512 104410          TYPED
3353 020514 104402 020520          TYPE   .+2          ;TYPE STACK IN DECIMAL
3354 020522 000207          RTS    PC           ;RETURN
3355
3356          ;ADD 4K TO TEST ADDR
3357
3358          ;XYZ*****?*****
3359          ;*****
3360          ;*****
3361          ;XYZ*****?*****
3362          ;*****
3363          ;*****
3364          ;XYZ*****?*****
3365          ;*****
3366 020524 005237 001230          .UPDAT: INC     WORK1      ;INC BANK #
3367 020530 062737 020000 001144    ADD     #20000,BUF    ;UPDATE BY 4K
3368 020536 103005          BCC     IS          ;BRANCH IF NO OVERFLOW
3369 020540 005237 001026          INC     TEMP3        ;INCREMENT FOR RSBAE
3370 020544 013777 001026 160310    MOV     TEMP3,RSBAE
3371 020552 000002          IS:     RTI
3372
3373 020554 020556          .SBTTL .TYPED          ;TYPED = TRAP+40 (104440)
3374          ;$TYPED - CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3375
3376 020556 012737 100040 021004    .TYPED: MOV     #100040,.DSIGN ;SET BLANK SWITCH AND SIGN
3377 020564 010046          MOV     R0,-(6)      ;PUSH R0 ON STACK
3378 020566 010146          MOV     R1,-(6)      ;PUSH R1 ON STACK
3379 020570 010246          MOV     R2,-(6)      ;PUSH R2 ON STACK
3380 020572 010346          MOV     R3,-(6)      ;PUSH R3 ON STACK
3381 020574 010546          MOV     R5,-(6)      ;PUSH R5 ON STACK
3382 020576 012737 100040 021004    MOV     #100040,.DSIGN ;SET BLANK SWITCH AND SIGN
3383 020604 016605 000016          MOV     16(6),R5     ;GET DATA TO BE TYPED
3384 020610 100004          BPL     IS          ;BR IF INPUT IS POS.
3385 020612 005405          NEG     R5          ;MAKE THE BINARY NUMBER POS.
3386 020614 112737 000055 021004    MOV     #'-.DSIGN    ;MAKE THE ASCII NUMBER NEG.
3387 020622 005000          IS:     CLR     R0          ;ZERO THE CONSTANTS INDEX
3388 020624 012703 020774          MOV     #.DBLK,R3    ;SETUP THE OUTPUT POINTER
```

3389	020630	112723	000040		MOVB	#' ,(R3)+	;SET THE FIRST CHARACTER TO A BLANK
3390	020634	005002		2\$:	CLR	R2	;CLEAR THE BCD NUMBER
3391	020636	016001	020764		MOV	.DTBL(R0),R1	;GET THE CONSTANT
3392	020642	160105		3\$:	SUB	R1,R5	;FORM THIS BCD DIGIT
3393	020644	002402			BLT	4\$;BR IF DONE
3394	020646	005202			INC	R2	;INCREASE THE BCD DIGIT BY 1
3395	020650	000774			BR	3\$	
3396	020652	060105		4\$:	ADD	R1,R5	;ADD BACK THE CONSTANT
3397	020654	005702			TST	R2	;CHECK IF BCD DIGIT=0
3398	020656	001003			BNE	5\$	FALL THROUGH IF 0
3399	020660	105737	021005		TSTB	.DSIGN+1	STILL DOING LEADING 0'S?
3400	020664	100410			BMI	7\$	BR IF YES
3401	020666	106337	021005		ASLB	.DSIGN+1	MSD?
3402	020672	103003			BCC	6\$	BR IF NO
3403	020674	113763	021004	177777	MOVB	.DSIGN,-1(R3)	YES--SET THE SIGN
3404	020702	052702	000060		BIS	#'0,R2	MAKE THE BCD DIGIT ASCII
3405	020706	052702	000040		BIS	#',R2	MAKE IT A SPACE IF NOT ALREADY A DIGIT
3406	020712	110223			MOVB	R2,(R3)+	PUT THIS CHARACTER IN THE OUTPUT BUFFER
3407	020714	005720			TST	(R0)+	JUST INCREMENTING
3408	020716	020027	000010		CMF	R0,#10	CHECK THE TABLE INDEX
3409	020722	002744			BLT	2\$	GO DO THE NEXT DIGIT
3410	020724	003002			BGT	8\$	GO TO EXIT
3411	020726	010502			MOV	R5,R2	GET THE LSD
3412	020730	000764			BR	6\$	GO CHANGE TO ASCII
3413	020732	105013			CLRB	(R3)	SET THE TERMINATOR
3414	020734	012605			MOV	(6)+,R5	POP STACK INTO R5
3415	020736	012603			MOV	(6)+,R3	POP STACK INTO R3
3416	020740	012602			MOV	(6)+,R2	POP STACK INTO R2
3417	020742	012601			MOV	(6)+,R1	POP STACK INTO R1
3418	020744	012600			MOV	(6)+,R0	POP STACK INTO R0
3419	020746	016666	000002	000004	MOV	2(6),4(6)	FUDGE DATA
3420	020754	012616			MOV	(6)+,(6)	OFF STACK
3421	020756	104402	020774		TYPE	..DBLK	NOW TYPE THE NUMBER
3422	020762	000002			RTI		RETURN
3423							
3424	020764	023420	001750	000144	.DTBL:	10000.,1000.,100.,10.	
3425	020772	000012			.DBLK:	.BLKW 4	
3426	020774	000004			.DSIGN:	0	
3427	021004	000000					
3428							
3429		000001			.END		

K07

CNPLP1	010640	2126	2128	2140*														
CMPY	003264	1106	1130	1149	1178	1184	1190	1202*										
CONDAR	007306	1865	1867*															
COMERR	001214	933*	2621	2641*	2778*	2779												
COMPAR	010444	1429	1622	2100*														
CONN	002220	1000	1071*															
CRLFLF	000757	794*	1042	1284	1611	2323	2785	2829	3257									
DA =	000004	841*	1250	1264	1340	1398	1443	1567	1599	1638	2122	2402	2454	2470				
		2578	2727															
DATA	000620	769*	1338	1376	1440													
DATAT	003710	1200	1296*	1493														
DATTES	002426	1107	1110*	1115														
DB =	000002	840*	1398	1599	1638	2454	2470	2526	2578	2742								
DISBUF	007234	1252	1289	1351	1410	1454	1856*	2457	2474	2530	2580							
DISPLA=	177570	722*	1954*	2922*	2926*													
DKCMD =	104416	1242	1257	1327	1361	1424	1553	1581	1619	2450	2465	2522	2571	2574				
		3137*	3309															
DKINT	006452	1728*	1836															
DMA	001140	911*	987*	1226*	1238	1272	1278	1312*	1539*	1540*	1664*	1686	1762*	1861*				
		1862	1870	1875	1886*	1900	2225	2446*	2461*	2517*	2562*	3296*						
DONE	013774	2629	2634	2646*														
DONEE	002204	1055	1065*	1671														
DOWN	012646	2447	2485*	2488														
DROP	001130	869*	1013	2417*	2588	2625	2781*	2786										
DRP	015002	2681	2770	2772*														
DRVENO	001620	1004*	1196	3291														
DS =	000040	844*	1340	1443	1567	1738	2122	2402	2737	3337								
DSKRD	004570	1420	1422*	1447														
DT =	000240	848*	2749															
DVNUM	001660	1009	1012*															
ELH	004634	1430	1432*															
ERCLR =	104414	1235	1255	1322	1354	1422	1551	1579	1617	3136*								
ERCOUN	001152	916*	2100*	2235*														
ERRCL	012322	965	1103	2410*														
ERRORS	001002	804*	2941*	2961														
ERRVEC=	000004	944*	997*	998*														
ERTAB	017310	2411	2637	2678*	3213*													
ESH	004524	1413*	1503															
ESH1	004520	1358	1412*															
EXGEN	010230	2030	2046*															
EXGEN1	010366	2081*																
EXMFLG	003242	1195	1197*	1670	2854													
EXRAX	006126	1573	1605	1626	1644*													
EXRXX	006140	1645	1647*															
EXTDR	007576	1892	1898	1921*														
EXTMEM	011514	981	1100	2282*														
EXTPP	006236	1668	1671*															
EXTPPR	006204	1500	1664*	2795														
EXTT	011472	2265	2270*															
FILCHR	001014	808*	2875	2876														
FILDAT	010010	1975*	1978															
FLAG	001132	908*	957*	992*	995*	1099*	1102*	1104*	1108*	1151*	1171*	1180*	1186*	1192*				
		1194	1197*	1198	1224*	1248*	1262*	1292*	1319*	1324	1330	1341*	1342	1350*				
		1356	1363	1414	1418*	1427	1432	1456	1497	1523*	1541*	1555	1569*	1570				
		1583	1613*	1625	1652*	1667	1676	1679*	1712	1728*	1748*	1760	1856	1867				
		1884	1888*	1889	1893	1920*	1959	2238	2240	2452	2468	2524	2560*	2576				

STAEUF	006616	1071	1090	1764*																	
STAEV	001074	855#	1073*	1086	1310	1810	2282	3293*													
STATUS	001070	850#	1316*	1534*																	
STBCOM	001100	857#	1093*																		
STMP2	011672	2267	2326#																		
STTEST	001752	107	1020#	1052																	
SWITCH	001200	927#	2113*	2131	2135*	2137*															
SWR =	177570	721#	722	968	990	993	1016	1038	1368	1370	1381	1425	1434	1495							
		1557	1585	1620	1628	1630	1749	2091	2181	2190	2233	2311	2442	2584							
		2628	2684	2769	2772	2850	2902	2913	2942	2952	2955										
SWROCT	001150	915#	985	989	1101	1144*	1145	1225	1229*	1233*	1291*	1311	1891	2052*							
		2053	2300*	2565*	2566																
SW0	= 000001	617#																			
SW1	= 000002	616#																			
SW10	= 002000	607#	2649	2941																	
SW11	= 004000	606#	2913																		
SW12	= 010000	605#	2649																		
SW13	= 020000	604#	2942																		
SW14	= 040000	603#	2902																		
SW15	= 100000	602#																			
SW2	= 000004	615#																			
SW3	= 000010	614#																			
SW4	= 002020	613#																			
SW5	= 000040	612#																			
SW6	= 000100	611#																			
SW7	= 000200	610#																			
SW8	= 000400	609#	2902																		
SW9	= 001000	608#	2955																		
TAG	004630	1426	1428	1431#																	
TAG1	006020	1621	1624#																		
TOMA	001146	914#	1161*	1762																	
TEMP1	001022	812#	1087	1475	1478	3330*															
TEMP2	001024	813#	1222*	1309*	1522*	1665*	1695	1818	1973	2025	2040	2150	2167	2253							
		2438*	2509*	3227	3246																
TEMP3	001026	814#	1310*	1474*	1475	1477*	1478	1480*	1482	1697	1820	3248	3307*	3369*							
		3370																			
TEMP4	001030	815#	1699*	1700*	1701																
TIEOUT	017450	997	3257#																		
TIMES	016034	2917	2932#																		
TIMUP	007176	1844#	3043																		
TPB	001020	810#	2876*	2881*																	
TPS	001016	809#	2877	2882																	
TRACK	001136	910#																			
TRLERR	006534	1735	1737	1748#																	
TRY	001640	1007#	1011																		
TRYNX	002130	1021	1031	1037	1049#	1669															
TST1	003310	1220#																			
TST2	003710	1300#																			
TST3	005166	1513#																			
TST4	006202	1663#																			
TYPE =	104402	1018	1042	1046	1060	1071	1074	1090	1091	1094	1105	1111	1129	1135							
		1148	1152	1163	1177	1183	1189	1202	1284	1338	1339	1375	1377	1391							
		1395	1440	1441	1565	1566	1592	1593	1596	1603	1608	1611	1636	1637							
		1790	1881	2088	2121	2195	2199	2229	2232	2275	2320	2323	2371	2401							
		2439	2478	2536	2582	2600	2605	2606	2607	2610	2611	2612	2615	2616							
		2617	2620	2623	2630	2633	2689	2692	2696	2700	2704	2707	2708	2711							

F08

MAINDEC-11-DEK-B-A RH11-RSQ3 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 98
DERS88.P11 27-OCT-76 11:03 CROSS REFERENCE TABLE -- USER SYMBOLS

.TYPE0	016170	2970#	3132
.TYPES	016200	2972#	3133
.UPDAT	020524	3140	3366#

END	1#	596#														
LDPDR	2325#	2356	2357	2358	2359	2360	2361	2362	2363							
NEWST	1#	596#	1217	1297	1510	1660										
OOT11	1#															
OOT11X	1#															
POP	1#	748#	2908	2911	3003	3034	3113	3175	3414							
PRINT	1#	748#	1018	1091	1105	1111	1129	1135	1148	1152	1163	1177	1183	1189	1391	
	1395	1596	1881	2088	2195	2199	2229	2232	2478	2536	2582	2600	2605	2607	2610	
	2612	2615	2617	2620	2623	2630	2633	2689	2692	2696	2700	2704	2707	2708	2711	
	2714	2719	2724	2729	2734	2739	2744	2751	2756	2761	2766	2771	2782	2827	2830	
	2833	2836	2841	2944	2950	3042	3180	3200	3258	3331	3335	3353				
PUSH	1#	748#	2905	2974	3020	3057	3066	3153	3377							
SCOPE.	1#	1217#	1297#	1510#	1660#											
SET	3117#	3130	3131	3132	3133	3134	3135	3136	3137	3138	3139	3140	3141	3142	3143	
	3144	3145	3373													
TYPADR	1#															
TYPBIT	1#	748#														
TYPCHR	1#	748#														
TYPDEC	1#	748#	3351													
TYPLIN	1#	748#														
TYPOCS	1#	748#	1043	1058	1882	2230	2601	2608	2613	2618	2621	2631	2767	2783		
TYPOCT	1#	748#	1392	1396	1597	2690	2693	2697	2701	2705	2709	2712	2720	2725	2730	
	2735	2740	2745	2752	2757	2762	2948									
TYPTXT	1#	748#	1018	1091	1105	1110	1128	1134	1147	1152	1162	1176	1182	1188	1390	
	1395	1596	1881	2087	2195	2199	2229	2232	2478	2536	2582	2599	2605	2607	2610	
	2612	2615	2617	2620	2623	2630	2633	2689	2692	2695	2699	2703	2707	2708	2711	
	2714	2719	2724	2729	2734	2739	2744	2751	2756	2761	2766	2771	2782	2826	2830	
	2833	2836	2841	3258	3331	3334	3353									
SCATCH	1#	596#	618													
SCHTAG	1#	596#	801													
SDONE	1#	596#	2644													
SEQUAT	1#	596#	717													
SHLT	1#	596#	2933													
SKMMR	1#	596#	674													
SOCTAL	1#	596#	2962													
SPOWER	1#	596#	3010													
SRAND2	1#															
SRAND4	1#															
SRODEC	1#															
SRDLIN	1#	596#	3183													
SRODOCT	1#	596#	3146													
SSCOPE	1#	596#	2895													
SSET	3117#	3130	3131	3132	3133	3134	3135	3136	3137	3138	3139	3140	3141	3142	3143	
	3144	3145	3373													
SSETUP	1#	596#	948													
SSMR	1#															
SSDOOC	1#	596#	599													
STRAP	1#	596#	3117													
STYPE	1#	596#	2855													
STYPEA	1#	596#	3050													
STYPED	1#	596#	3374													
SJMMR	1#															

H08

MAINDEC-11-DERSB-A RH11-RS03 DATA AND RELIABILITY TEST MACY11 27(1006) 27-OCT-76 11:08 PAGE 101
DERSBB.P11 27-OCT-76 11:03 CROSS REFERENCE TABLE -- MACRO NAMES

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

NOW. SEQ/SCL/CRF/NL: TOC=DERSBB.SML, DERSBB.P11
RUN-TIME: 26 40 3 SECONDS
RUN-TIME RATIO: 3586/71=50.3
CORE USED: 22K (43 PAGES)

