

RK611  
RK06, RK07

RK611 DSKLS PRT 1  
CZR6AD0

AH-9339D-MC  
FICHE 1 OF 2

MAR 1982  
COPYRIGHT © 76-81  
MADE IN USA





RK611  
RK06, RK07

RK611 DSKLS PRT 1  
CZR6AD0

AH-9339D-MC  
FICHE 2 OF 2

MAR 1982  
COPYRIGHT © 76-81  
MADE IN USA





.REM %

IDENTIFICATION

PRODUCT CODE: AC-9338D-MC  
PRODUCT NAME: CZR6AD0 RK611 DSKLS PRT1  
DATE: AUGUST 10 1981  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: BRIAN LE BLANC

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENCE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976,1981 BY DIGITAL EQUIPMENT CORPORATION

31  
30  
29  
28  
27  
26  
25  
24  
23  
22  
21  
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1



TABLE OF CONTENTS

32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87

1.0 ABSTRACT

2.0 REQUIREMENTS  
2.1 EQUIPMENT  
2.2 PRELIMINARY PROGRAMS

3.0 OPERATING PROGRAMS  
3.1 LOADING PROCEDURE  
3.2 STARTING PROCEDURE  
3.3 OPTIONAL SWITCH SETTING  
3.4 RUN TIME

4.0 OPERATING PROCEDURES

5.0 PROGRAM DESCRIPTION

6.0 ERROR REPORTING

1.0 ABSTRACT

THE RK611 DISKLESS CONTROLLER DIAGNOSTIC: PART 1 READS AND WRITES EVERY RK611 REGISTER, TESTS THE INTERRUPT MECHANISM, AND TEST THE SILO LOADING LOGIC. NO RK06 DRIVE IS REQUIRED FOR PROGRAM EXECUTION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP-11 SYSTEM (16K CORE MEMORY)  
CONSOLE TERMINAL  
DECTAPE, PAPER TAPE READER, OR DECDISK  
RK611 CONTROLLER

2.2 PRELIMINARY PROGRAMS

NONE

3.0 OPERATING PROCEDURES

3.1 LOADING PROCEDURE

THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING ABSOLUTE LOADER OR FROM XXDP MEDIA SUPPORTED BY XXDP.

3.2 STARTING PROCEDURE

LOCATION 200 - START PROGRAM



88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
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  
140  
141  
142  
143

LOCATION 204 - RESTART PROGRAM

LOCATION 214 - REQUEST BUS ADDRESS, VECTOR ADDRESS, AND  
PRIORITY MODIFICATION

### 3.3 OPTIONAL SWITCH SETTINGS

SW15 - HALT PROGRAM  
SW14 - LOOP ON TEST  
SW13 - INHIBIT ERROR TYPE OUT  
SW12 - ABORT AFTER 20 ERRORS  
SW11 - INHIBIT ITERATION COUNT  
SW10 - BELL ON ERROR  
SW9 - LOOP ON ERROR  
SW8 - LOOP ON TEST IN SWITCHES 0-7

### 3.5 RUN TIME

FIRST PASS 7 SECONDS  
SUBSEQUENT PASSES 2 MINUTES

### 4.0 OPERATING PROCEDURES

THE PROGRAM IS EXECUTED BY STARTING AT THE APPROPRIATE ADDRESS.

### 5.0 PROGRAM DESCRIPTION

#### TEST 1 ADDRESS ALL RK611 REGISTERS

THIS TEST WILL ACCESS ALL RK611 REGISTERS AND CHECK TO  
MAKE SURE THAT NON-EXISTENT MEMORY DOES NOT OCCUR.  
A NON-EXISTENT MEMORY INDICATES EITHER THAT THE RK611  
REGISTER BASE ADDRESS IS INCORRECT OR THAT THE  
RK611 DOES NOT RESPOND TO UNIBUS DIALOGUE.

#### \*\*RESET, CONTROLLER CLEAR, AND TRI-STATE TESTS

#### TEST 2 RESET RK611 AND VERIFY REGISTERS

RESET THE RK611 CONTROLLER AND READ ALL REGISTER OF THE  
RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT  
THEY ARE CORRECT. REEXAMINE COMMAND AND STATUS REGISTER  
TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE  
COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE  
DID NOT SET.

THE SUCCESSFUL EXECUTION OF THIS TEST VERIFIES THAT NO  
BIT OF THE TRI-STATE BUS IS STUCK TO ONE.

#### TEST 3 CONTROLLER CLEAR AND VERIFY REGISTERS

INITIALIZE THE RK611 CONTROLLER WITH A CONTROLLER



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  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199

CLEAR AND READ ALL REGISTER OF THE RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE CORRECT. REEXAMINE COMMAND AND STATUS REGISTER 1 TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE DID NOT SET.

TEST 4 WRITE BUS ADDRESS WITH 177777 (PART 1)

THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777 AND CHECK IF EQUAL TO 177776 AND THAT NO REGISTER INTERA TAKES PLACE. A RESET IS DONE AT THE END OF THE TEST TO MAKE SURE THE BUS ADDRESS CLEARS AND ALL RK611 REGISTERS ARE IN THEIR INITIALIZED STATE.

TEST 5 WRITE BUS ADDRESS WITH 177777 (PART 2)

THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777. A CONTROLLER CLEAR IS DONE. MAKE SURE THE BUS ADDRESS CL

TEST 6 WRITE WORD COUNT REG. WITH 177777

THIS TEST WILL WRITE THE WORD COUNT REGISTER TO 0 AND 177777 AND CHECK IF EQUAL TO 0 AND 177777 RESPECTIVELY AND THAT NO REGISTER INTERACTION TAKES PLAC ISSUE A CONTROLLER CLEAR AND MADE SURE THAT THE WORD COUNT REGISTER DOES NOT CHANGE.

TEST 7 WRITE DISK ADDRESS WITH 177777

THIS TEST WILL WRITE THE DISK ADDRESS REGISTER TO 177777 AND CHECK IF EQUAL TO 003437 AND THAT NO REGISTER INTERA TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.

\*\*REGISTER INTERACTION TESTS

ALL REGISTER INTERACTION TESTS CONSISTS OF WRITING A REGISTER AND CHECKING ITS CONTENTS AGAINST EXPECTED CONTENTS. THEN ALL OTHER REGISTERS ARE READ EXCEPT THE DATA BUFFER TO CHECK WHETHER THEY HAVE CHANGED FROM THEIR INITIALIZED CONDITIONS.

TEST 10 REGISTER INTERACTION USING BUS ADDRESS (PART 1)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE CLEAR TO HE RK611 CONTROLLER. IT WILL THEN WRITE THE WO COUNT REGISTER TO 0.

THE TEST ITSELF WILL CONSIST OF WRITING THE



200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255

BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A  
TEST IF BUS ADDRESS IS CORRECT AND THAT NO  
REGISTER INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 11 REGISTER INTERACTION USING BUS ADDRESS (PART 2)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE  
CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO  
COUNT REGISTER TO 0.

THE TEST ITSELF WILL CONSIST OF WRITING THE  
BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A  
TEST IF BUS ADDRESS IS CORRECT AND THAT NO  
REGISTER INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE  
CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO  
COUNT REGISTER TO 0.

THE TEST ITSELF WILL CONSIST OF WRITING THE  
BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A  
TEST IF BUS ADDRESS IS CORRECT AND THAT NO  
REGISTER INTERACTION TAKES PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLE  
CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WO  
COUNT REGISTER TO 0.

THE TEST ITSELF WILL CONSIST OF WRITING THE  
BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS A  
TEST IF BUS ADDRESS IS CORRECT AND THAT NO  
REGISTER INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	



170000 177400 177760 177777

TEST 14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO REGISTER INTERACTION TAKES PLACE.

000000 000020 000400 010000  
000002 000040 001000 020000  
000004 000100 002000 040000  
000010 000200 004000

TEST 15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO REGISTER INTERACTION TAKES PLACE.

077776 077756 077376 067776  
077774 077736 076776 057776  
077772 077676 075776 037776  
077766 077576 073776

TEST 16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO REGISTER INTERACTION TAKES PLACE.

000002 000076 001776 037776  
000006 000176 003776 077776  
000016 000376 007776 000000  
000036 000776 017776

TEST 17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)

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  
300  
301  
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  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO REGISTER INTERACTION TAKES PLACE.

000000	074000	077600	077770
040000	076000	077700	077774
060000	077000	077740	077775
070000	077400	077760	

TEST 20 REGISTER INTERACTION USING SPARE REG

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE NO INTERACTION TAKES PLACE.

TEST 21 REGISTER INTERACTION USING WORD COUNT (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 22 REGISTER INTERACTION USING WORD COUNT (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 23 REGISTER INTERACTION USING WORD COUNT (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE PLACE.



368  
369  
370  
371  
372  
373  
374  
375  
376  
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

PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 24 REGISTER INTERACTION USING WORD COUNT (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS  
WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS  
AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

TEST 25 REGISTER INTERACTION USING DISK ADDRESS (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS  
WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH  
THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND  
SURE NO INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS  
WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH  
THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND  
SURE NO INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 27 REGISTER INTERACTION USING DISK ADDRESS (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS  
WRITE THE WORD COUNT REGISTER WITH 0.



424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479

WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND SURE NO INTERACTION TAKES PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTER WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND SURE NO INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

TEST 31 REGISTER INTERACTION USING ATTN/OFFSET (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTER WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS MAKE SURE NO INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 32 REGISTER INTERACTION USING ATTN/OFFSET (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTER WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS MAKE SURE NO INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTER WRITE THE WORD COUNT REGISTER WITH 0.



480  
481  
482  
483  
484  
485  
486  
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  
529  
530  
531  
532  
533  
534  
535

WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS. MAKE SURE NO INTERACTION TAKES PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 34 REGISTER INTERACTION USING ATTN/OFFSET (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTER. WRITE THE WORD COUNT REGISTER WITH 0.

WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS. MAKE SURE NO INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

TEST 35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1)

RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR. WRITE THE WORD COUNT REGISTER WITH ZERO.

WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING COMMANDS SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS REGISTER 2 AND CHECK FOR REGISTER INTERACTION.

000000	000010	000400	010000
000001	000020	001000	020000
000002	000100	002000	040000
000004	000200	004000	100000

TEST 36 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2)

RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR. WRITE THE WORD COUNT REGISTER WITH ZERO.

WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING COMMANDS SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS REGISTER 2 AND CHECK FOR REGISTER INTERACTION.

177737	177727	177337	167737
177736	177717	176737	157737
177735	177637	175737	137737
177733	177537	173737	077737

TEST 37 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3)



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  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591

RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR  
WRITE THE WORD COUNT REGISTER WITH ZERO.

WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C  
SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST  
REG. 2 AND CHECK FOR REGISTER INTERACTION.

000001	000037	001737	037737
000003	000137	005737	077737
000007	000337	007737	177737
000017	000737	017737	000000

TEST 40 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4)

RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.  
WRITE THE WORD COUNT REGISTER WITH ZERO.

WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING C  
SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND ST  
REG. 2 AND CHECK FOR REGISTER INTERACTION.

100000	174000	177600	177734
140000	176000	177700	177736
160000	177000	177720	177737
170000	177400	177730	000000

TEST 41 CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS

THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR T  
INITIALIZE THE BUS ADDRESS REGISTER AND COMMAND  
AND STATUS REGISTER 1. IT WILL ALSO VERIFY THAT ALL  
OTHER REGISTERS REMAIN IN THE INITIALIZED STATE.

TEST 42 REGISTER INTERACTION USING DRIVE STATUS

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE WORD COUNT TO 0, WRITE DRIVE STATUS REGISTER  
WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO  
INTERACTION TAKES PLACE.

TEST 43 REGISTER INTERACTION USING ERROR REGISTER

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGIS  
WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO  
INTERACTION TAKES PLACE.

TEST 44 REGISTER INTERACTION USING MAINT REG 2

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
REGISTER 2 TO 177777 AND NO INTERACTION TAKES PLACE.



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  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647

TEST 45 REGISTER INTERACTION USING MAINT REG 3

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO, WRITE MAINTENANCE  
REGISTER 3 TO 177777 AND NO INTERACTION TAKES PLACE.

TEST 46 REGISTER INTERACTION WITH DISK CYLINDER (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 47 REGISTER INTERACTION WITH DISK CYLINDER (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 50 REGISTER INTERACTION WITH DISK CYLINDER (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000001	000037	000777	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 51 REGISTER INTERACTION WITH DISK CYLINDER (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	



648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703

TEST 52 REGISTER INTERACTION USING MAINT REG 1 (PART 1)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 53 REGISTER INTERACTION USING MAINT REG 1 (PART 2)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

177777	177767	177577	173777	077777
177776	177757	177377	167777	
177775	177737	176777	157777	
177773	177677	175777	137777	

TEST 54 REGISTER INTERACTION USING MAINT REG 1 (PART 3)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

000001	000037	000177	017777	000000
000003	000077	001777	037777	
000007	000177	003777	077777	
000017	000377	007777	177777	

TEST 55 REGISTER INTERACTION USING MAINT REG 1 (PART 4)

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF C  
ARE CORRECT AND NO INTERACTION TAKES PLACE.

100000	174000	177600	177770	000000
140000	176000	177700	177774	
160000	177000	177740	177776	
170000	177400	177760	177777	

TEST 56 REGISTER INTERACTION WITH PATTERN REG.



704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTERN  
REGISTER TO 177777 AND MAKE SURE IT REMAINS 0 AND  
NO INTERACTION TAKES PLACE.

TEST 57 REGISTER INTERACTION WITH POSITION REG.

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE POSITION  
REGISTER TO 177777 AND MAKE SURE IT STAYS AT THE  
INITIALIZED CONDITION AND NO INTERACTION TAKES PLACE.

\*\*INTERRUPT TESTS

TEST 60 RK611 INTERRUPT

STORE LOCATIONS 0-776, LOAD LOCATIONS 0-776 TO TRAP ALL  
POSSIBLE INTERRUPTS. LOWER PROCESSOR PRIORITY TO ZERO.  
MAKE SURE THAT NO INTERRUPT OCCURS. NOW SET INTERRUPT  
ENABLE AND READY. VERIFY THAT THE INTERRUPT OCCURS AT  
PROPER VECTOR ADDRESS. MAKE SURE THAT INTERRUPT IS  
CLEARED AFTER IT IS GIVEN.

TEST 61 INTERRUPT PRIORITY

SET UP PRIORITY TO 1 LESS THAN INTERRUPT PRIORITY.  
WRITE READY WITH INTERRUPT ENABLE. MAKE SURE INTERRUPT  
NOW SET UP PRIORITY EQUAL TO INTERRUPT PRIORITY.  
WRITE INTERRUPT ENABLE WITH READY. MAKE SURE INTERRUPT  
DOES NOT OCCUR. NOW LOWER PRIORITY AND MAKE  
INTERRUPT HAS BEEN STORED.

TEST 62 SETTING INTERRUPT ENABLE

CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK611  
INTERRUPTS BY SETTING PROCESSOR PRIORITY TO ZERO.  
SET INTERRUPT ENABLE AND MAKE SURE NO INTERRUPTS OCCUR.

TEST 63 INTERRUPT CLEARING

SET UP PRIORITY TO SEVEN. CREATE INTERRUPT BY SETTING  
INTERRUPT ENABLE READY. AND CLEAR IT WITH CONTROLLER  
CLEAR. SET INTERRUPT ENABLE. NOW LOWER PRIORITY  
TO MAKE SURE NO INTERRUPT OCCURS.

\*\*SILO TESTS

TEST 64 READ SILO WHEN EMPTY



760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815

READ SILO WHEN EMPTY. CHECK FOR DATA LATE AND CONTROLLE  
ERROR. ISSUE CONTROLLER CLEAR AND CHECK IF ERROR RESET.

TEST 65 SILO LOADING AND UNLOADING OF ONE WORD

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 CONTROLLER.  
CLEAR WORD COUNT REGISTER.

WRITE A WORD OF 177777 INTO THE SILO. CHECK ALL OTHER  
REGISTERS FOR INTERACTION PROBLEMS. CHECK THAT OUTPUT  
READY IS SET IN COMMAND AND STATUS REGISTER 2. IF NOT  
WAIT A REASONABLE TIME FOR IT.

IF OUTPUT READY COMES UP IN A REASONABLE TIME, READ BACK  
CONTENTS AND MAKE SURE IT IS 177777. CHECK FOR NO CONTR  
ERROR, NO DATA LATE, INPUT READY SET, OUTPUT READY RESET  
NOW READ ANOTHER WORD FROM THE SILO TO MAKE SURE DATA  
LATE AND CONTROLLER ERROR SET.

TEST 66 ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)

ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO  
ZERO.

WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERAC  
PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND  
STATUS REGISTER 2. IF NOT WAIT.

IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK  
IF CORRECT.

THE FOLLOWING CONFIGURATIONS ARE USED:

000000	000010	000200	004000	100000
000001	000020	000400	010000	
000002	000040	001000	020000	
000004	000100	002000	040000	

TEST 67 ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)

ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO  
ZERO.

WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERAC  
PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND  
STATUS REGISTER 2. IF NOT WAIT.

IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK  
IF CORRECT.

THE FOLLOWING CONFIGURATIONS ARE USED:

177777	177767	177577	173777	077777
177776	177757	177377	167777	



816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871

177775 177737 176777 157777  
177773 177677 175777 137777

TEST 70 ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)

ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO ZERO.

WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERAC PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND STATUS REGISTER 2. IF NOT WAIT.

IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK IF CORRECT.

THE FOLLOWING CONFIGURATIONS ARE USED:

000001 000037 000777 017777 000000  
000003 000077 001777 037777  
000007 000177 003777 077777  
000017 000377 007777 177777

TEST 71 ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)

ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO ZERO.

WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERAC PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND STATUS REGISTER 2. IF NOT WAIT.

IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK IF CORRECT.

THE FOLLOWING CONFIGURATIONS ARE USED:

100000 174000 177600 177770 000000  
140000 176000 177700 177774  
160000 177000 177740 177776  
170000 177400 177760 177777

TEST 72 SILO FILL

THIS TEST WILL WRITE THE SILO WITH 66 DIFFERENT PATTERNS CHECK INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH WORD WRITTEN. IT WILL THEN READ ALL 66 WORDS BACK CHECK CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR E WORD READ. AN EXTRA READ IS THEN DONE TO MAKE SURE THE SILO IS EMPTY.

TEST 73 SILO CAPACITY DATA LATE



872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927

WRITE 67 WORDS IN THE SILO AND MAKE SURE DATA LATE ONLY OCCURS ON THE 67TH WORD. CLEAR RK611 WITH CONTROLLER CL CHECK INPUT READY AND OUTPUT READY FOR INITIALIZED STATE

TEST 74 INTERRUPT DUE TO DATA LATE

ALLOW RK611 INTERRUPTS. SET INTERRUPT ENABLE. NOW READ ONE WORD FROM DATA BUFFER AND MAKE SURE THAT DATA LATE CAUSES INTERRUPT. BEFORE CLEARING ERROR ALLOW RK611 INTERRUPTS AND MAKE SURE IT DOES NOT OCCUR AGAIN. NOW CLEAR CONTROLLER WITH A CONTROLLER CLEAR.

TEST 75 INTERRUPT CLEARING AND DATA LATE

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. CREATE A CONTROLLER ERROR (DATA LATE) BY READING THE DATA BUFFER WHEN EMPTY. CLEAR RK611 CONTROLLER WITH A CONTROLLER CL SET INTERRUPT ENABLE AND LOWER PROCESSOR PRIORITY.

MAKE SURE AN INTERRUPT DOES NOT OCCUR.

TEST 76 INTERRUPT ENABLE AND DATA LATE

CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. ALLOW RK611 INTERRUPTS. READ DATA BUFFER TO GENERATE INTERRUPT PENDING. MAKE SURE INTERRUPT DOES NOT OCCUR.

NOW SET INTERRUPT ENABLE AND MAKE SURE INTERRUPTS OCCURS

6.0 ERROR REPORTING

THE GENERAL FORMAT OF ERROR REPORTS IS:

OPERATION DESCRIPTION AND ERROR DESCRIPTION

TEST	ERROR		
NUM	PC		
XXXXXX	YYYYYY		
EXPECT	ACTUAL	OTHER	PERTENANT
REG	REG		INFORMATION
ZZZZZZ	WWWWWW	AAAAAA	

NOTE: MORE THAN ONE SET OF EXPECT/ACTUAL REGISTERS MAY BE PRINTED OUT. OTHER PERTENANT INFORMATION MAY CONSIST OF MORE THAN ONE WORD.

[ END OF DOCUMENT ]

%



928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983

001100

000011  
000012  
000015  
000200  
177776  
177774  
177772  
177570  
177570

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
000006  
000007

000000

```
*** REV 003 ***
.TITLE CZR6AD0 RK611 DSKLS CTRL PRT1
.*COPYRIGHT (C) 1976,1981
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY ROY SPITZER
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
.*
.SBTTL OPERATIONAL SWITCH SETTINGS
.*
.*          SWITCH          USE
.*          -----          -
.*          15          HALT ON ERROR
.*          14          LOOP ON TEST
.*          13          INHIBIT ERROR TYPEOUTS
.*          12          ABORT PROGRAM AFTER 20 ERRORS
.*          11          INHIBIT ITERATIONS
.*          10          BELL ON ERROR
.*          9           LOOP ON ERROR
.*          8           LOOP ON TEST IN SWR<7:0>
.SBTTL BASIC DEFINITIONS
.*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR          ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE          ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
HT= 11          ;;CODE FOR HORIZONTAL TAB
LF= 12          ;;CODE FOR LINE FEED
CR= 15          ;;CODE FOR CARRIAGE RETURN
CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776     ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774  ;;STACK LIMIT REGISTER
PIRQ= 177772   ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570   ;;HARDWARE SWITCH REGISTER
DDISP= 177570  ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0          ;;GENERAL REGISTER
R1= %1          ;;GENERAL REGISTER
R2= %2          ;;GENERAL REGISTER
R3= %3          ;;GENERAL REGISTER
R4= %4          ;;GENERAL REGISTER
R5= %5          ;;GENERAL REGISTER
R6= %6          ;;GENERAL REGISTER
R7= %7          ;;GENERAL REGISTER
SP= %6          ;;STACK POINTER
PC= %7          ;;PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS
PRO= 0          ;;PRIORITY LEVEL 0
```



984	000040	PR1= 40	::PRIORITY LEVEL 1
985	000100	PR2= 100	::PRIORITY LEVEL 2
986	000140	PR3= 140	::PRIORITY LEVEL 3
987	000200	PR4= 200	::PRIORITY LEVEL 4
988	000240	PR5= 240	::PRIORITY LEVEL 5
989	000300	PR6= 300	::PRIORITY LEVEL 6
990	000340	PR7= 340	::PRIORITY LEVEL 7
991			
992		;*"SWITCH REGISTER" SWITCH DEFINITIONS	
993	100000	SW15= 100000	
994	040000	SW14= 40000	
995	020000	SW13= 20000	
996	010000	SW12= 10000	
997	004000	SW11= 4000	
998	002000	SW10= 2000	
999	001000	SW09= 1000	
1000	000400	SW08= 400	
1001	000200	SW07= 200	
1002	000100	SW06= 100	
1003	000040	SW05= 40	
1004	000020	SW04= 20	
1005	000010	SW03= 10	
1006	000004	SW02= 4	
1007	000002	SW01= 2	
1008	000001	SW00= 1	
1009		.EQUIV SW09,SW9	
1010		.EQUIV SW08,SW8	
1011		.EQUIV SW07,SW7	
1012		.EQUIV SW06,SW6	
1013		.EQUIV SW05,SW5	
1014		.EQUIV SW04,SW4	
1015		.EQUIV SW03,SW3	
1016		.EQUIV SW02,SW2	
1017		.EQUIV SW01,SW1	
1018		.EQUIV SW00,SW0	
1019			
1020		;*DATA BIT DEFINITIONS (BIT00 TO BIT15)	
1021	100000	BIT15= 100000	
1022	040000	BIT14= 40000	
1023	020000	BIT13= 20000	
1024	010000	BIT12= 10000	
1025	004000	BIT11= 4000	
1026	002000	BIT10= 2000	
1027	001000	BIT09= 1000	
1028	000400	BIT08= 400	
1029	000200	BIT07= 200	
1030	000100	BIT06= 100	
1031	000040	BIT05= 40	
1032	000020	BIT04= 20	
1033	000010	BIT03= 10	
1034	000004	BIT02= 4	
1035	000002	BIT01= 2	
1036	000001	BIT00= 1	
1037		.EQUIV BIT09,BIT9	
1038		.EQUIV BIT08,BIT8	
1039		.EQUIV BIT07,BIT7	



```
1040 .EQUIV BIT06,BIT6
1041 .EQUIV BIT05,BIT5
1042 .EQUIV BIT04,BIT4
1043 .EQUIV BIT03,BIT3
1044 .EQUIV BIT02,BIT2
1045 .EQUIV BIT01,BIT1
1046 .EQUIV BIT00,BIT0
1047
1048 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1049 000004 ERRVEC= 4 ;:TIME OUT AND OTHER ERRORS
1050 000010 RESVEC= 10 ;:RESERVED AND ILLEGAL INSTRUCTIONS
1051 000014 TBITVEC=14 ;: "T" BIT
1052 000014 TRTVEC= 14 ;:TRACE TRAP
1053 000014 BPTVEC= 14 ;:BREAKPOINT TRAP (BPT)
1054 000020 IOTVEC= 20 ;:INPUT/OUTPUT TRAP (IOT) **SCOPE**
1055 000024 PWRVEC= 24 ;:POWER FAIL
1056 000030 EMTVEC= 30 ;:EMULATOR TRAP (EMT) **ERROR**
1057 000034 TRAPVEC=34 ;: "TRAP" TRAP
1058 000060 TKVEC= 60 ;:TTY KEYBOARD VECTOR
1059 000064 TPVEC= 64 ;:TTY PRINTER VECTOR
1060 000240 PIRQVEC=240 ;:PROGRAM INTERRUPT REQUEST VECTOR
1061 000114 MEMVEC= 114 ;:MEMORY PARITY VECTOR
1062 172100 MEMBAS= 172100 ;:MEMORY PARITY BASE ADDRESS
1063 000001 PAR.EN= 1 ;:ALLOW MEMORY PARITY OPTION
1064 120210 AVECT1= 120210 ;:DEFINE RK611 VECTOR ADDRESS
1065 000005 APRIOR= 5 ;:DEFINE RK611 PRIORITY
1066 177440 ABASE= 177440 ;:DEFINE BASE OF RK611 REGISTERS
1067
1068 .SBTTL RK611 CONTROLLER REGISTER DEFINITION
1069
1070 000000 RKCS1= 0 ;:CONTROL AND STATUS REGISTER 1
1071 000002 RKWC= 2 ;:WORD COUNT REGISTER
1072 000004 RKBA= 4 ;:BUS ADDRESS REGISTER
1073 000006 RKDA= 6 ;:DESIRED TRACK SECTOR REGISTER
1074 000010 RKCS2= 10 ;:CONTROL AND STATUS REGISTER 2
1075 000012 RKDS= 12 ;:DRIVE STATUS REGISTER
1076 000014 RKER= 14 ;:ERROR REGISTER
1077 000016 RKASOF= 16 ;:ATTENTION SUMMARY AND OFFSET REGISTER
1078 000020 RKDCYL= 20 ;:DESIRED CYLINDER REGISTER
1079 000024 RKDB= 24 ;:DATA BUFFER
1080 000026 RKMR1= 26 ;:MAINTENANCE REGISTER 1
1081 000034 RKMR2= 34 ;:MAINTENANCE REGISTER 2
1082 000036 RKMR3= 36 ;:MAINTENANCE REGISTER 3
1083 000030 RKECPS= 30 ;:ECC POSITION INFORMATION
1084 000032 RKECPT= 32 ;:ECC PATTERN INFORMATION
1085 000022 RKSPAR= 22 ;:SPARE REGISTER
1086
1087 .SBTTL DRIVE COMMANDS
1088
1089 000001 SELDRV= 01 ;:SELECT DRIVE
1090 000003 PACK= 03 ;:PACK ACKNOWLEDGE
1091 000005 CLEAR= 05 ;:DRIVE CLEAR
1092 000007 UNLOAD= 07 ;:UNLOAD
1093 000011 SRTSPL= 11 ;:START SPINDLE
1094 000013 RECAL= 13 ;:RECALIBRATE
1095 000015 OFFSET= 15 ;:OFFSET
```



DRIVE COMMANDS

1096	000017	SEEK= 17	:SEEK
1097	000021	RDDATA= 21	:READ DATA
1098	000023	WRDATA= 23	:WRITE DATA
1099	000025	RDHEAD= 25	:READ HEADER
1100	000027	WRHEAD= 27	:WRITE HEADER AND DATA
1101	000031	WRTCHK= 31	:WRITE CHECK
1102	000300	INTR= 300	:GENERATE INTERRUPT TO CPU

.SBTTL CONTROL AND STATUS REGISTER 1 BITS

1106	000001	GO= BIT0	:GO BIT
1107	000100	IE= BIT6	:INTERRUPT ENABLE
1108	000200	RDY= BIT7	:CONTROLLER READY
1109	000400	BA16= BIT8	:BUS ADDRESS BIT 16
1110	001000	BA17= BIT9	:BUS ADDRESS BIT 17
1111	002000	CDT= BIT10	:CONTROLLER DRIVE TYPE (0=RK06)
1112	004000	CTO= BIT11	:CONTROLLER TIMED OUT WAITING FOR : DRIVE RESPONSE
1114	010000	CFMT= BIT12	:CONTROLLER DRIVE FORMAT (0=26 SECTOR, 1=24 SECTOR)
1115	020000	SPAR= BIT13	:DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1116	040000	DI= BIT14	:DRIVE INTERRUPT



```
1117      100000      CERR= BIT15      ;CONTROLLER ERROR
1118      100000      CCLR= BIT15      ;CONTROLLER CLEAR
1119
1120      .SBTTL CONTROL AND STATUS REGISTER 2 BITS
1121
1122      000007      DRVMSK= 7      ;MASK FOR DRIVE SELECTION CODE
1123      000010      RLS= BIT3      ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1124      000020      BAI= BIT4      ;BUS ADDRESS INCREMENT INHIBIT
1125      000040      SCLR= BIT5      ;CLEAR CONTROLLER AND ALL DRIVES
1126      000100      IR= BIT6      ;INPUT READY
1127      000200      OR= BIT7      ;OUTPUT READY
1128      000400      UFE= BIT8      ;UNIT FIELD ERROR
1129      001000      MDS= BIT9      ;MULTIPLE DRIVE SELECT
1130      002000      PGE= BIT10     ;PROGRAMMING ERROR
1131      004000      NEM= BIT11     ;NON-EXISTENT MEMORY
1132      010000      NED= BIT12     ;NON-EXISTENT DRIVE
1133      020000      UPE= BIT13     ;UNIBUS PARITY ERROR
1134      040000      WCE= BIT14     ;WRITE CHECK ERROR
1135      100000      DLT= BIT15     ;DATA LATE ERROR
1136
1137      .SBTTL ERROR REGISTER BIT DEFINITION
1138
1139      000001      ILF= BIT0      ;ILLEGAL FUNCTION CODE
1140      000002      SKI= BIT1      ;SEEK INCOMPLETE
1141      000004      NXF= BIT2      ;NON-EXECUTABLE DRIVE FUNCTION
1142      000010      DRPAR= BIT3     ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1143      000020      FMTE= BIT4     ;FORMAT ERROR
1144      000040      DTYE= BIT5     ;DRIVE TYPE ERROR
1145      000100      ECH= BIT6     ;ECC HARD
1146      000200      BSE= BIT7     ;BAD SECTOR ERROR
1147      000400      HVRC= BIT8     ;HEADER VRC ERROR
1148      001000      COE= BIT9     ;CYLINDER ADDRESS OVERFLOW ERROR
1149      002000      IDAE= BIT10    ;INVALID DISK ADDRESS ERROR
1150      004000      WLE= BIT11    ;WRITE LOCK ERROR
1151      010000      DTE= BIT12    ;DRIVE TIMING ERROR
1152      020000      OPI= BIT13    ;OPERATION (SEARCH) INCOMPLETE
1153      040000      UNS= BIT14    ;DRIVE UNSAFE
1154      100000      DCK= BIT15    ;DATA CHECK
1155
1156      .SBTTL STATUS REGISTER BIT DEFINITION
1157
1158      000001      DRA= BIT0      ;DRIVE AVAILABLE (CONTROLLER IS SET IF
1159      ; THIS BIT IS RESET)
1160      000004      OFST= BIT2     ;DRIVE OFFSET
1161      000010      ACLO= BIT3     ;AC LOW
1162      000020      SPDLSS= BIT4   ;SPEED LOSS
1163      000040      DROT= BIT5     ;DRIVE OFF TRACK
1164      000100      VV= BIT6      ;VOLUME VALID
1165      000200      DRDY= BIT7     ;DRIVE READY
1166      000400      DDT= BIT8     ;DRIVE TYPE (0=RK06)
1167      004000      WRL= BIT11    ;WRITE LOCK
1168      020000      PIP= BIT13    ;POSITIONING IN PROGRESS
1169      040000      DSC= BIT14    ;DRIVE STATUS CHANGE
1170      100000      SVAL= BIT15   ;STATUS VALID
1171
1172      .SBTTL MAINTENANCE REGISTER 1 BIT DEFINITION
```



```
1173
1174      000017      MESMSK= 17      ;MESSAGE MASK
1175
1176      000020      PAT= BIT4      ;FORCE EVEN PARITY ON DRIVE MESSAGE LINES
1177      000040      DMD= BIT5      ;DIAGNOSTIC MODE
1178      000100      MSP= BIT6      ;MAINTENANCE SECTOR PULSE
1179      000200      MIND= BIT7      ;MAINTENANCE INDEX
1180      000400      MCLK= BIT8      ;MAINTENANCE CLOCK
1181      001000      MERD= BIT9      ;MAINTENANCE ENCODED READ DATA
1182      002000      MEWD= BIT10     ;MAINTENANCE ENCODED WRITE DATA
1183      004000      PCA= BIT11     ;PRECOMPENSATION ADVANCE
1184      010000      PCD= BIT12     ;PRECOMPENSATION DELAY
1185      020000      ECCW= BIT13     ;ECC WORD IS BEING READ OR WRITTEN
1186      040000      WRTGAT= BIT14   ;WRITE GATE
1187      100000      RDGATE= BIT15   ;READ GATE
1188
1189      .SBTTL TRANSMITTED MESSAGE A
1190
1191      000020      S.SEK= BIT4      ;SEEK COMMAND
1192      000040      S.RECL= BIT5     ;RECALIBRATE COMMAND
1193      000100      S.STSP= BIT6     ;START SPINDLE COMMAND
1194      000200      S.RTC= BIT7     ;DRIVE RETURN TO CENTERLINE COMMAND
1195      000400      S.CLR= BIT8     ;CLEAR ERROR AND DSC
1196      001000      S.FMT= BIT9     ;FORMAT
1197      002000      S.UNLD= BIT10    ;UNLOAD
1198      004000      S.PACK= BIT11    ;SET VOLUME VALID (PACK ACKNOWLEDGE)
1199      .SBT.L TRAP CATCHER
1200
1201      .=0
1202      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1203      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1204      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1205      .=174
1206 000174 000000  DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
1207 000176 000000  SWREG: .WORD 0      ;;SOFTWARE SWITCH REGISTER
1208      .SBTTL STARTING ADDRESS(ES)
1209 000200 000137 002050  JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
1210 000204 000137 002040  JMP RESTRT  ;JUMP TO RESTART ROUTINE
1211      .=214
1212 000214 000137 002030  JMP PARM    ;JUMP TO OPERATOR ASSIGNED PARMETERS
1213      .SBTTL ACT11 HOOKS
1214
1215      ;*****
1216      ;HOOKS REQUIRED BY ACT11
1217      $SVPC=.      ;SAVE PC
1218      .=46
1219 000046 055244  $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1220      .=52
1221 000052 000000  .WORD 0      ;;2)SET LOC.52 TO ZERO
1222      .=$SVPC      ;;RESTORE PC
1223      .=1000
1224      .SBTTL APT PARAMETER BLOCK
1225
1226      ;*****
1227      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1228      ;*****
```



CZR6AD0 RK611  
CZR6AD.P11

KLS CTRL PRT1  
-SEP-81 13:43

MACY11 30(1046) 14-SEP-81 15:04 L 2  
APT PARAMETER BLOCK PAGE 25

SEQ 0024

1229 001000  
1230 00024  
1231 000 00200  
1232 00044  
1233 001000  
1234 001000  
1235  
1236  
1237  
1238  
1239 001000  
1240 001000 000000  
1241 001002 001214  
1242 001004 000001  
1243 001006 000007  
1244 001010 000007  
1245 001012 000032

. \$X= . ::SAVE CURRENT LOCATION  
. =24 ::SET POWER FAIL TO POINT TO START OF PROGRAM  
200 ::FOR APT START UP  
. =44 ::POINT TO APT INDIRECT ADDRESS PNTR.  
\$APTHDR ::POINT TO APT HEADER BLOCK  
. =. \$X ::RESET LOCATION COUNTER  
\*\*\*\*\*  
: SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
: INTERFACE SPEC.  
\$APTHD:  
\$HIBTS: .WORD 0 ::TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
\$MBADR: .WORD \$MAIL ::ADDRESS OF APT MAILBOX (BITS 0-15)  
\$STMT: .WORD 1 ::RUN TIM OF LONGEST TEST  
\$PASTM: .WORD 7 ::RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
\$UNITM: .WORD 7 ::ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
.WORD \$ETEND-\$MAIL/2 ::LENGTH MAILBOX-ETABLE(WORDS)



1246  
1247  
1248  
1249  
1250  
1251  
1252 001100  
1253 001100 001100  
1254 001100 000000  
1255 001102 000  
1256 001103 000  
1257 001104 000000  
1258 001106 000000  
1259 001110 000000  
1260 001112 000000  
1261 001114 000  
1262 001115 001  
1263 001116 000000  
1264 001120 000000  
1265 001122 000000  
1266 001124 000000  
1267 001126 000000  
1268 001130 000000  
1269 001132 000000  
1270 001134 000  
1271 001135 000  
1272 001136 000000  
1273 001140 177570  
1274 001142 177570  
1275 001144 177560  
1276 001146 177562  
1277 001150 177564  
1278 001152 177566  
1279 001154 000  
1280 001155 002  
1281 001156 012  
1282 001157 000  
1283 001160 000000  
1284 001162 000000  
1285 001164 000000  
1286 001166 000000  
1287 001170 000000  
1288 001172 000000  
1289 001174 000000  
1290 001176 000000  
1291 001200 000000  
1292 001202 000000  
1293 001204 177607 000377  
1294 001210 077  
1295 001211 015  
1296 001212 000012  
1297  
1298  
1299  
1300  
1301

.SBTTL COMMON TAGS

::\*\*\*\*\*  
:\*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS  
:\*USED IN THE PROGRAM.

SCMTAG: . =1100  
\$STNM: .WORD 0  
\$ERFLG: .BYTE 0  
\$ICNT: .WORD 0  
\$LPADR: .WORD 0  
\$LPERR: .WORD 0  
\$ERTTL: .WORD 0  
\$ITEMB: .BYTE 0  
\$ERMAX: .BYTE 1  
\$ERRPC: .WORD 0  
\$GDADR: .WORD 0  
\$BDADR: .WORD 0  
\$GDDAT: .WORD 0  
\$BDDAT: .WORD 0  
\$AUTOB: .BYTE 0  
\$INTAG: .BYTE 0  
\$SWR: .WORD DSWR  
\$DISPLAY: .WORD DDISP  
\$TKS: 177560  
\$TKB: 177562  
\$TPS: 177564  
\$TPB: 177566  
\$NULL: .BYTE 0  
\$FILLS: .BYTE 2  
\$FILLC: .BYTE 12  
\$TPFLG: .BYTE 0  
\$TMP0: .WORD 0  
\$TMP1: .WORD 0  
\$TMP2: .WORD 0  
\$TMP3: .WORD 0  
\$TMP4: .WORD 0  
\$TMP5: .WORD 0  
\$TMP6: .WORD 0  
\$TMP7: .WORD 0  
\$TIMES: 0  
\$ESCAPE: 0  
\$BELL: .ASCIZ <207><377><377>  
\$QUES: .ASCII /?/  
\$CRLF: .ASCII <15>  
\$LF: .ASCIZ <12>

:::START OF COMMON TAGS  
:::CONTAINS THE TEST NUMBER  
:::CONTAINS ERROR FLAG  
:::CONTAINS SUBTEST ITERATION COUNT  
:::CONTAINS SCOPE LOOP ADDRESS  
:::CONTAINS SCOPE RETURN FOR ERRORS  
:::CONTAINS TOTAL ERRORS DETECTED  
:::CONTAINS ITEM CONTROL BYTE  
:::CONTAINS MAX. ERRORS PER TEST  
:::CONTAINS PC OF LAST ERROR INSTRUCTION  
:::CONTAINS ADDRESS OF 'GOOD' DATA  
:::CONTAINS ADDRESS OF 'BAD' DATA  
:::CONTAINS 'GOOD' DATA  
:::CONTAINS 'BAD' DATA  
:::RESERVED--NOT TO BE USED  
:::AUTOMATIC MODE INDICATOR  
:::INTERRUPT MODE INDICATOR  
:::ADDRESS OF SWITCH REGISTER  
:::ADDRESS OF DISPLAY REGISTER  
:::TTY KBD STATUS  
:::TTY KBD BUFFER  
:::TTY PRINTER STATUS REG. ADDRESS  
:::TTY PRINTER BUFFER REG. ADDRESS  
:::CONTAINS NULL CHARACTER FOR FILLS  
:::CONTAINS # OF FILLER CHARACTERS REQUIRED  
:::INSERT FILL CHARS. AFTER A 'LINE FEED'  
:::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::USER DEFINED  
:::MAX. NUMBER OF ITERATIONS  
:::ESCAPE ON ERROR ADDRESS  
:::CODE FOR BELL  
:::QUESTION MARK  
:::CARRIAGE RETURN  
:::LINE FEED

.SBTTL APT MAILBOX-ETABLE

::\*\*\*\*\*  
:EVEN



1302	001214		\$MAIL:		::APT MAILBOX
1303	001214	000000	\$MSGTY:	.WORD	AMSGTY ::MESSAGE TYPE CODE
1304	001216	000000	\$FATAL:	.WORD	AFATAL ::FATAL ERROR NUMBER
1305	001220	000000	\$TESTN:	.WORD	ATESTN ::TEST NUMBER
1306	001222	000000	\$PASS:	.WORD	APASS ::PASS COUNT
1307	001224	000000	\$DEVCT:	.WORD	ADEVCT ::DEVICE COUNT
1308	001226	000000	\$UNIT:	.WORD	AUNIT ::I/O UNIT NUMBER
1309	001230	000000	\$MSGAD:	.WORD	AMSGAD ::MESSAGE ADDRESS
1310	001232	000000	\$MSGLG:	.WORD	AMSGLG ::MESSAGE LENGTH
1311	001234		\$ETABLE:		::APT ENVIRONMENT TABLE
1312	001234	000	\$ENV:	.BYTE	AENV ::ENVIRONMENT BYTE
1313	001235	000	\$ENVM:	.BYTE	AENVM ::ENVIRONMENT MODE BITS
1314	001236	000000	\$SWREG:	.WORD	ASWREG ::APT SWITCH REGISTER
1315	001240	000000	\$USWR:	.WORD	AUSWR ::USER SWITCHES
1316	001242	000000	\$CPUOP:	.WORD	ACPUOP ::CPU TYPE,OPTIONS
1317			*		BITS 15-11=CPU TYPE
1318			*		11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1319			*		11/70=06,PDQ=07,Q=10
1320			*		BIT 10=REAL TIME CLOCK
1321			*		BIT 9=FLOATING POINT PROCESSOR
1322			*		BIT 8=MEMORY MANAGEMENT
1323	001244	000	\$MAMS1:	.BYTE	AMAMS1 ::HIGH ADDRESS,M.S. BYTE
1324	001245	000	\$MTYP1:	.BYTE	AMTYP1 ::MEM. TYPE,BLK#1
1325			*		MEM.TYPE BYTE -- (HIGH BYTE)
1326			*		900 NSEC CORE=001
1327			*		300 NSEC BIPOLAR=002
1328			*		500 NSEC MOS=003
1329	001246	000000	\$MADR1:	.WORD	AMADR1 ::HIGH ADDRESS,BLK#1
1330			*		MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1331	001250	000	\$MAMS2:	.BYTE	AMAMS2 ::HIGH ADDRESS,M.S. BYTE
1332	001251	000	\$MTYP2:	.BYTE	AMTYP2 ::MEM. TYPE,BLK#2
1333	001252	000000	\$MADR2:	.WORD	AMADR2 ::MEM.LAST ADDRESS,BLK#2
1334	001254	000	\$MAMS3:	.BYTE	AMAMS3 ::HIGH ADDRESS,M.S.BYTE
1335	001255	000	\$MTYP3:	.BYTE	AMTYP3 ::MEM. TYPE,BLK#3
1336	001256	000000	\$MADR3:	.WORD	AMADR3 ::MEM.LAST ADDRESS,BLK#3
1337	001260	000	\$MAMS4:	.BYTE	AMAMS4 ::HIGH ADDRESS,M.S.BYTE
1338	001261	000	\$MTYP4:	.BYTE	AMTYP4 ::MEM. TYPE,BLK#4
1339	001262	000000	\$MADR4:	.WORD	AMADR4 ::MEM.LAST ADDRESS,BLK#4
1340	001264	120210	\$VECT1:	.WORD	AVECT1 ::INTERRUPT VECTOR#1,BUS PRIORITY#1
1341	001266	000000	\$VECT2:	.WORD	AVECT2 ::INTERRUPT VECTOR#2BUS PRIORITY#2
1342	001270	177440	\$BASE:	.WORD	ABASE ::BASE ADDRESS OF EQUIPMENT UNDER TEST
1343	001272	000000	\$DEVN:	.WORD	ADEVN ::DEVICE MAP
1344	001274	000000	\$CDW1:	.WORD	ACDW1 ::CONTROLLER DESCRIPTION WORD#1
1345	001276	000000	\$CDW2:	.WORD	ACDW2 ::CONTROLLER DESCRIPTION WORD#2
1346	001300		\$ETEND:		
1347			.MEXIT		



1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362 001300  
1363  
1364 001300 063333  
1365 001302 000000  
1366 001304 061500  
1367 001306 061644  
1368  
1369 001310 000000  
1370 001312 000000  
1371 001314 061506  
1372 001316 061664  
1373  
1374 001320 000000  
1375 001322 000000  
1376 001324 061516  
1377 001326 061710  
1378  
1379 001330 000000  
1380 001332 000000  
1381 001334 061530  
1382 001336 061734  
1383  
1384 001340 065050  
1385 001342 000000  
1386 001344 061542  
1387 001346 061760  
1388  
1389 001350 065142  
1390 001352 000000  
1391 001354 061546  
1392 001356 061774  
1393  
1394 001360 065222  
1395 001362 000000  
1396 001364 061546  
1397 001366 061774  
1398  
1399 001370 065260  
1400 001372 000000  
1401 001374 061546  
1402 001376 061774  
1403

.SBTTL ERROR POINTER TABLE

;\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
;\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
;\*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
;\*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).  
;\*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;\* EM ;;POINTS TO THE ERROR MESSAGE  
;\* DH ;;POINTS TO THE DATA HEADER  
;\* DT ;;POINTS TO THE DATA  
;\* DF ;;POINTS TO THE DATA FORMAT

\$ERRTB:  
; ERROR 1: NEM ACCESSING RK611 REGISTER  
EM1  
0  
DT001  
DF001  
; ERROR 2: ATTEMPTING TO INITIALIZE RK611  
EM2N: 0  
0  
DT002  
DF002  
; ERROR 3: ATTEMPTING TO WRITE REG  
EM3N: 0  
0  
DT003  
DF003  
; ERROR 4: ATTEMPTING TO CLEAR RK611  
EM4N: 0  
0  
DT004  
DF004  
; ERROR 5: INTERRUPT DID NOT OCCUR  
EM24  
0  
DT005  
DF005  
; ERROR 6: UNEXPECTED INTERRUPT WHEN LOWERING PROCESSOR PRIORITY  
EM25  
0  
DT006  
DF006  
; ERROR 7: VECTOR ADDRESS INCORRECT  
EM26  
0  
DT006  
DF006  
; ERROR 10: INTERRUPT DID NOT CLEAR IN RK611  
EM27  
0  
DT006  
DF006  
; ERROR 11: EXPECTED INTERRUPT DID NOT OCCUR AT PROCESSOR PRIORITY



1404	001400	065314	EM28	
1405	001402	000000	0	
1406	001404	061556	DT011	
1407	001406	062020	DF011	
1408			ERROR 12:	UNEXPECTED INTERRUPT OCCURRED AT PROCESSOR PRIORITY
1409	001410	065376	EM29	
1410	001412	000000	0	
1411	001414	061556	DT011	
1412	001416	062020	DF011	
1413			ERROR 13:	INTERRUPT DID NOT OCCUR WHEN PRIORITY LOWERED
1414	001420	065455	EM30	
1415	001422	000000	0	
1416	001424	061556	DT011	
1417	001426	062020	DF011	
1418			ERROR 14:	SETTENG INTERRUPT ENABLE CAUSED EXPECTED INTERRUPT
1419	001430	065526	EM31	
1420	001432	000000	0	
1421	001434	061542	DT005	
1422	001436	061760	DF005	
1423			ERROR 15:	CONTROLLER CLEAR DID NOT CLEAR STORED INTERRUPT
1424	001440	065577	EM32	
1425	001442	000000	0	
1426	001444	061542	DT005	
1427	001446	061760	DF005	
1428			ERROR 16:	ATTEMPTING TO READ SILO WHEN EMPTY - CS2 INCORRECT
1429	001450	065714	EM34	
1430	001452	067154	EM1009	
1431	001454	061564	DT016	
1432	001456	062040	DF016	
1433			ERROR 17:	ATTEMPTING TO READ SILO WHEN EMPTY - CS1 INCORRECT
1434	001460	065714	EM34	
1435	001462	066656	EM1000	
1436	001464	061564	DT016	
1437	001466	062040	DF016	
1438			ERROR 20:	ATTEMPTING TO CLEAR DATA LATE - CS1 INCORRECT
1439	001470	065757	EM35	
1440	001472	066656	EM1000	
1441	001474	061564	DT016	
1442	001476	062040	DF016	
1443			ERROR 21:	ATTEMPTING TO CLEAR DATA LATE - CS2 INCORRECT
1444	001500	065757	EM35	
1445	001502	066766	EM1004	
1446	001504	061564	DT016	
1447	001506	062040	DF016	
1448			ERROR 22:	ATTEMPTING TO READ SILO CONTAINING ONE WORD CS1 INCORRECT
1449				
1450	001510	066015	EM36	
1451	001512	066656	EM1000	
1452	001514	061564	DT016	
1453	001516	062040	DF016	
1454			ERROR 23:	ATTEMPTING TO READ SILO CONTAINING ONE WORD CS2 INCORRECT
1455				
1456	001520	066015	EM36	
1457	001522	066766	EM1004	
1458	001524	061564	DT016	
1459	001526	062040	DF016	



1460			:	ERROR 24:	ATTEMPTING TO LOAD SILO - CS1 INCORRECT
1461	001530	066071		EM37	
1462	001532	066656		EM1000	
1463	001534	061600		DT024	
1464	001536	062064		DF024	
1465			:	ERROR 25:	ATTEMPTING TO LOAD SILO - CS2 INCORRECT
1466	001540	066071		EM37	
1467	001542	066766		EM1004	
1468	001544	061600		DT024	
1469	001546	062064		DF024	
1470			:	ERROR 26:	ATTEMPTING TO READ SILO - CS1 INCORRECT
1471	001550	066121		EM38	
1472	001552	066656		EM1000	
1473	001554	061616		DT026	
1474					
1475	001556	062110		DF026	
1476			:	ERROR 27:	ATTEMPTING TO READ SILO - CS2 INCORRECT
1477	001560	066121		EM38	
1478	001562	066766		EM1004	
1479	001564	061616		DT026	
1480	001566	062110		DF026	
1481			:	ERROR 30:	ATTEMPTING TO READ SILO - DATA INCORRECT
1482	001570	066121		EM38	
1483	001572	067126		EM1008	
1484	001574	061616		DT026	
1485	001576	062110		DF026	
1486			:	ERROR 31:	ATTEMPTING TO LOAD FULL SILO - CS2 INCORRECT
1487	001600	066151		EM39	
1488	001602	066766		EM1004	
1489	001604	061564		DT016	
1490	001606	062040		DF016	
1491			:	ERROR 32:	ATTEMPTING TO LOAD FULL SILO - CS1 INCORRECT
1492	001610	065376		EM29	
1493	001612	066656		EM1000	
1494	001614	061564		DT016	
1495	001616	062040		DF016	
1496			:	ERROR 33:	DATA LATE DID NOT CAUSE EXPECTED INTERRUPT
1497	001620	066221		EM40	
1498	001622	000000		0	
1499	001624	061542		DT005	
1500	001626	061760		DF005	
1501			:	ERROR 34:	UNEXPECTED INTERRUPT DUE TO UNCLEARED CONTROLLER ERROR
1502	001630	066267		EM41	
1503	001632	000000		0	
1504	001634	061542		DT005	
1505	001636	061760		DF005	
1506			:	ERROR 35:	CONTROLLER CLEAR DID NOT CLEAR PENDING INTERRUPT DUE TO CONTROLLER ERROR
1507			:		
1508	001640	066351		EM42	
1509	001642	000000		0	
1510	001644	061542		DT005	
1511	001646	061760		DF005	
1512			:	ERROR 36:	CONTROLLER ERROR CAUSED INTERRUPT WITH INTERRUPT ENABLE RESET
1513			:		
1514	001650	066456		EM43	
1515	001652	000000		0	



1516	001654	061542	DT005	
1517	001656	061760	DF005	
1518			ERROR 37:	INTERRUPT DID NOT OCCUR WHEN INTERRUPT ENABLE
1519			:	SET WITH INTERRUPT PENDING DUE TO DATA LATE
1520	001660	066542	EM44	
1521	001662	062333	DH000A	
1522	001664	061546	DT006	
1523	001666	061774	DF006	
1524			ERROR 40:	UNEXPECTED MEMORY PARITY ENABLE TRAP
1525	001670	063266	EM000	
1526	001672	062364	DH000C	
1527	001674	061474	DT000	
1528	001676	061640	DF000	



```
1529 .SBTTL TEMPORARY STORAGE FOR RK611 CONTROLLER REGISTER
1530
1531 001700 000000 T.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
1532 001702 000000 T.WC: .WORD 0 ;WORD COUNT REGISTER
1533 001704 000000 T.BA: .WORD 0 ;BUS ADDRESS REGISTER
1534 001706 000000 T.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
1535 001710 000000 T.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
1536 001712 000000 T.DS: .WORD 0 ;DRIVE STATUS REGISTER
1537 001714 000000 T.ER: .WORD 0 ;ERROR REGISTER
1538 001716 000000 T.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
1539 001720 000000 T.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
1540 001722 000000 T.DB: .WORD 0 ;DATA BUFFER
1541 001724 000000 T.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
1542 001726 000000 T.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
1543 001730 000000 T.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
1544 001732 000000 T.ECPS: .WORD 0 ;ECC POSITION INFORMATION
1545 001734 000000 T.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
1546 001736 000000 T.SPAR: .WORD 0 ;SPARE REGISTER
1547
1548 .SBTTL EXPECTED RK611 CONTROLLER REGISTERS
1549
1550 001740 000000 E.CS1: .WORD 0 ;CONTROL AND STATUS REGISTER 1
1551 001742 000000 E.WC: .WORD 0 ;WORD COUNT REGISTER
1552 001744 000000 E.BA: .WORD 0 ;BUS ADDRESS REGISTER
1553 001746 000000 E.DA: .WORD 0 ;DESIRED TRACK SECTOR REGISTER
1554 001750 000000 E.CS2: .WORD 0 ;CONTROL AND STATUS REGISTER 2
1555 001752 000000 E.DS: .WORD 0 ;DRIVE STATUS REGISTER
1556 001754 000000 E.ER: .WORD 0 ;ERROR REGISTER
1557 001756 000000 E.ASOF: .WORD 0 ;ATTENTION SUMMARY AND OFFSET REGISTER
1558 001760 000000 E.DCYL: .WORD 0 ;DESIRED CYLINDER REGISTER
1559 001762 000000 E.DB: .WORD 0 ;DATA BUFFER
1560 001764 000000 E.MR1: .WORD 0 ;MAINTENANCE REGISTER 1
1561 001766 000000 E.MR2: .WORD 0 ;MAINTENANCE REGISTER 2
1562 001770 000000 E.MR3: .WORD 0 ;MAINTENANCE REGISTER 3
1563 001772 000000 E.ECPS: .WORD 0 ;ECC POSITION INFORMATION
1564 001774 000000 E.ECPT: .WORD 0 ;ECC PATTERN INFORMATION
1565 001776 000000 E.SPAR: .WORD 0 ;SPARE REGISTER
1566
1567 .SBTTL PROGRAM DEFINED VARIABLES
1568
1569 002000 000210 RKVEC: .WORD 210 ;RK611 VECTOR ADDRESS
1570 002002 000240 RKPRI: .WORD PR5 ;RK611 PRIORITY
1571 002004 000000 SRTFLG: .WORD 0 ;START FLAG
1572 ; 0 = 200
1573 ; 1 = 214
1574 ; -1 = 204
1575 002006 000000 ERRCNT: .WORD 0 ;ERROR COUNT FOR SWITCH 12 ABORT
1576 002010 000000 CONFIG: .WORD 0 ;DATA WRITTEN IN REGISTER
1577 002012 000000 CFG1: .WORD 0 ;DATA GENERATION WORD
1578 002014 000000 PREREG: .WORD 0 ;PREVIOUS REGISTER CONTENTS
1579 002016 000000 SAVFLG: .WORD 0 ;TRAP CATCHER SAVED
1580 002020 000000 PRIOR: .WORD 0 ;PROCESSOR PRIORITY
1581 002022 000000 SILCNT: .WORD 0 ;NUMBER OF WORDS READ OR WRITTEN IN SILO
1582 002024 000000 TRAPPC: .WORD 0 ;TRAP PC FOR UNEXPECTED MAIN MEMORY CHECK
1583 002026 000000 SAVSWR: .WORD 0 ;RESTORE SWITCH REGISTER
```



```
1584 .SBTTL PROGRAM SETUP
1585
1586 002030 012737 000001 002004 PARM: MOV #1,SRTFLG ;LOAD START FLAG FOR PARMETER START
1587 002036 000406 BR START1
1588
1589 002040 012737 177777 002004 RESTRT: MOV #-1,SRTFLG ;LOAD START FLAG FOR RESTART
1590 002046 000402 BR START1
1591
1592 002050 005037 002004 START: CLR SRTFLG ;CLEAR START FLAG
1593 002054 000005 START1: RESET ;RESET THE WHOLE SYSTEM
1594 002056 012706 001100 MOV #STACK,SP ;INITIALIZE STACK POINTER
1595 002062 012746 000340 MOV #PR7,-(SP) ;LOAD STACK TO LOCK OUT ALL INTERRUPTS
1596 002066 012746 002074 MOV #1$,-(SP) ;LOAD START OF PROGRAM
1597 002072 000002 RTI ;LOAD PSW
1598
1599 002074 1$:
1600 .SBTTL INITIALIZE THE COMMON TAGS
1601 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1602 002074 012706 001100 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
1603 002100 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
1604 002102 022706 001140 CMP #SWR,R6 ;;DONE?
1605 002106 001374 BNE -6 ;;LOOP BACK IF NO
1606 002110 012706 001100 MOV #STACK,SP ;;SETUP THE STACK POINTER
1607 ;;INITIALIZE A FEW VECTORS
1608 002114 012737 055416 000020 MOV #SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1609 002122 012737 000340 000022 MOV #340,@#IOTVEC+2 ;;LEVEL 7
1610 002130 012737 056416 000030 MOV #ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1611 002136 012737 000340 000032 MOV #340,@#EMTVEC+2 ;;LEVEL 7
1612 002144 012737 061404 000034 MOV #STRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1613 002152 012737 000340 000036 MOV #340,@#TRAPVEC+2 ;;LEVEL 7
1614 002160 012737 061254 000024 MOV #PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1615 002166 012737 000340 000026 MOV #340,@#PWRVEC+2 ;;LEVEL 7
1616 002174 013737 055110 055102 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
1617 002202 005037 001200 CLR $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
1618 002206 005037 001202 CLR $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1619 002212 112737 000001 001115 MOV #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
1620 002220 012737 002220 001106 MOV #,$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1621 002226 012737 002226 001110 MOV #,$LPERR ;;SETUP THE ERROR LOOP ADDRESS
1622 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1623 ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1624 002234 013746 000004 MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
1625 002240 012737 002274 000004 MOV #64$,@ERRVEC ;;SET UP ERROR VECTOR
1626 002246 012737 177570 001140 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
1627 002254 012737 177570 001142 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
1628 002262 022777 177777 176650 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
1629 002270 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1630 ;;AND THE HARDWARE SWR IS NOT = -1
1631 002272 000403 BR 65$ ;;BRANCH IF NO TIMEOUT
1632 002274 012716 002302 64$: MOV #65$,(SP) ;;SET UP FOR TRAP RETURN
1633 002300 000002 RTI
1634 002302 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
1635 002310 012737 000174 001142 MOV #DISPREG,DISPLAY
1636 002316 012637 000004 66$: MOV (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
1637
1638 002322 005037 001222 CLR $PASS ;;CLEAR PASS COUNT
1639 002326 132737 000200 001235 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
```



```

1640 002334 001403          BEQ      67$          ;;YES,USE NON-APT SWITCH
1641 002336 012737 001236 001140  MOV     #$$SWREG,SWR  ;;NO,USE APT SWITCH REGISTER
1642 002344          67$:
1643 002344 005037 002006          CLR     ERRCNT        ;CLEAR ERROR COUNT
1644          .SBTTL  TYPE PROGRAM NAME
1645          ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1646 002350 005227 177777          INC     #-1           ;;FIRST TIME?
1647 002354 001055          BNE     68$          ;;BRANCH IF NO
1648 002356 022737 055244 000042  CMP     #$$ENDAD,@#42 ;;ACT-11?
1649 002364 001451          BEQ     68$          ;;BRANCH IF YES
1650 002366 104401 002434          TYPE   ,69$         ;;TYPE ASCIZ STRING
1651          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1652 002372 005737 000042          TST    @#42         ;;ARE WE RUNNING UNDER XXDP/ACT?
1653 002376 001012          BNE     70$          ;;BRANCH IF YES
1654 002400 123727 001234 000001  CMPB   $ENV,#1      ;;ARE WE RUNNING UNDER APT?
1655 002406 001406          BEQ     70$          ;;BRANCH IF YES
1656 002410 023727 001140 000176  CMP     SWR,#SWREG  ;;SOFTWARE SWITCH REG SELECTED?
1657 002416 001005          BNE     71$          ;;BRANCH IF NO
1658 002420 104406          GTSWR                    ;;GET SOFT-SWR SETTINGS
1659 002422 000403          BR     71$
1660 002424 112737 000001 001134 70$:  MOVB   #1,$AUTOB    ;;SET AUTO-MODE INDICATOR
1661 002432          71$:
1662 002432 000426          BR     68$          ;;GET OVER THE ASCIZ
1663          ;;69$:
1664          .ASCIZ <CRLF>/RK611 DISKLESS DIAGNOSTIC: PART 1 CZR6AD0/<CRLF>
1665 002510          68$:
1666 002510 022737 000001 002004  CMP     #1,SRTFLG   ;CHECK IF PARAMETER START
1667 002516 001122          BNE     15$         ;NO, CONTINUE SETUP
1668 002520 104401 062134 5$:   TYPE   ,OPR001     ;TYPE 'RK611 BUS ADDRESS ( ) ='
1669 002524 013746 001270          MOV     $BASE,-(SP) ;SAVE $BASE FOR TYPEOUT
1670 002530 104402          TYPOC                    ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1671 002532 104401 062163          TYPE   ,OPR002
1672 002540 012637 001160          RDOCT                    ;GET VALUE
1673 002544 001407          MOV     (SP)+,$TMP0
1674 002546 022737 160000 001160  BEQ     7$          ;CHECK IF <CR>
1675 002554 101361          CMP     #160000,$TMP0 ;CHECK IF IN I/O PAGE
1676 002556 013737 001160 001270 7$:  MOV     $TMP0,$BASE  ;LOAD NEW BUS ADDRESS
1677 002564 104401 062171          TYPE   ,OPR003     ;TYPE 'RK611 VECTOR ADDRESS ( ) ='
1678 002570 013746 001264          MOV     $VECT1,-(SP) ;TYPE OUT CURRENT VECTOR ADDRESS
1679 002574 042716 160000          BIC     #160000,(SP)
1680 002600 104402          TYPOC
1681 002602 104401 062163          TYPE   ,OPR002
1682 002606 104412          RDOCT                    ;GET VALUE
1683 002610 012637 001160          MOV     (SP)+,$TMP0
1684 002614 001412          BEQ     10$         ;CHECK IF <CR>
1685 002616 022737 001000 001160  CMP     #1000,$TMP0  ;CHECK IF LEGAL
1686 002624 101757          BLOS   7$
1687 002626 042737 017777 001264  BIC     #17777,$VECT1 ;LOAD NEW VECTOR ADDRESS
1688 002634 053737 001160 001264  BIS     $TMP0,$VECT1
1689 002642 104401 062221 10$:  TYPE   ,OPR004     ;TYPE 'RK611 PRIORITY ( ) ='
1690 002646 005046          CLR    -(SP)        ;MAKE ROOM ON THE STACK
1691 002650 113716 001265          MOVB   $VECT1+1,(SP)
1692 002654 006216          ASR    (SP)         ;SHIFT 5 BITS RIGHT
1693 002656 006216          ASR    (SP)
1694 002660 006216          ASR    (SP)
1695 002662 006216          ASR    (SP)
  
```



1696	002664	006216			ASR	(SP)		
1697	002666	104402			TYPOC			
1698	002670	104401	062163		TYPE	.OPR002		
1699	002674	104412			RDOCT			;GET VALUE
1700	002676	012637	001160		MOV	(SP)+,\$TMP0		
1701	002702	001430			BEQ	15\$		;CHECK FOR DEFAULT
1702	002704	022737	000007	001160	CMP	#7,\$TMP0		;CHECK IF LEGAL
1703	002712	103753			BLO	10\$		
1704	002714	022737	000004	001160	CMP	#4,\$TMP0		
1705	002722	101347			BHI	10\$		
1706	002724	006337	001160		ASL	\$TMP0		;SHIFT 5 BITS LEFT
1707	002730	006337	001160		ASL	\$TMP0		
1708	002734	006337	001160		ASL	\$TMP0		
1709	002740	006337	001160		ASL	\$TMP0		
1710	002744	006337	001160		ASL	\$TMP0		
1711	002750	042737	160000	001264	BIC	#160000,\$VECT1		;STORE NEW PRIORITY
1712	002756	153737	001160	001265	BISB	\$TMP0,\$VECT1+1		
1713	002764	013737	001264	002000	15\$: MOV	\$VECT1,RKVEC		;STORE RK611 VECTOR
1714	002772	042737	160000	002000	BIC	#160000,RKVEC		;CLEAR PRIORITY BITS
1715	003000	113737	001265	002002	MOV	\$VECT1+1,RKPRI		;STORE PRIORITY
1716	003006	005737	002016		TST	SAVFLG		;CHECK IF TRAP CATCHER IS TO BE RESTORED
1717	003012	001412			BEQ	NEWPAS		;NO, GO TO FIRST TEST
1718	003014	012701	070036		MOV	#SAVVEC,R1		;RESTORE TRAP CATCHER
1719	003020	005000			CLR	R0		
1720	003022	012703	000400		MOV	#400,R3		;STORE COUNT
1721	003026	012120			16\$: MOV	(R1)+,(R0)+		
1722	003030	005303			DEC	R3		
1723	003032	001375			BNE	16\$		
1724	003034	005037	002016		CLR	SAVFLG		;INDICATE THAT TRAP CATCHER HAS BEEN RESTORED
1725								
1726	003040	004737	055264		NEWPAS: JSR	PC,CHKPAR		;CHECK FOR MEMORY CHECK ENABLE
1727	003044	013702	001270		MOV	\$BASE,R2		;LOAD RK611 BASE ADDR
1728	003050	012746	000340		MOV	#PR7,-(SP)		;LOCK OUT INTERRUPTS
1729	003054	012746	003062		MOV	#TST1,-(SP)		
1730	003060	000002			RTI			



1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741 003062 000004  
1742 003064 012737 000764 001200  
1743 003072 013702 001270  
1744 003076 012737 003130 000004  
1745 003104 012737 000340 000006  
1746 003112 012703 000020  
1747 003116 005712  
1748 003120 005722  
1749 003122 005303  
1750 003124 001374  
1751 003126 000405  
1752  
1753 003130 062706 000004  
1754 003134 010237 001122  
1755 003140 104001  
1756 003142 012737 000006 000004  
1757 003150 005037 000006  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776 003154 000004  
1777 003156 012737 000012 001200  
1778 003164 013702 001270  
1779 003170 012737 063410 001310  
1780 003176 000005  
1781 003200 004737 055264  
1782 003204 016237 000000 001126  
1783 003212 022737 000200 001126  
1784 003220 001407  
1785 003222 012737 000200 001124  
1786 003230 012737 066656 001312

```
*****  
:TEST 1 ADDRESS ALL RK611 REGISTERS  
:  
:* THIS TEST WILL ACCESS ALL RK611 REGISTERS AND CHECK TO  
:* MAKE SURE THAT NON-EXISTENT MEMORY DOES NOT OCCUR.  
:* A NON-EXISTENT MEMORY INDICATES EITHER THAT THE RK611  
:* REGISTER BASE ADDRESS IS INCORRECT OR THAT THE  
:* RK611 DOES NOT RESPOND TO UNIBUS DIALOGUE.  
:*****
```

```
TST1: SCOPE  
MOV #500, $TIMES ;:DO 500. ITERATIONS  
MOV $BASE, R2 ;:LOAD RK611 BASE  
MOV #10$, ERRVEC ;:LOAD VECTOR FOR NEM  
MOV #PR7, ERRVEC+2  
MOV #16., R3 ;:LOAD NUMBER OF REGISTER  
5$: TST (R2) ;:ADDRESS RK611 REGISTER  
TST (R2)+ ;:INCREMENT TO NEXT REGISTER  
DEC R3 ;:CHECK IF ALL REGISTERS ADDRESS  
BNE 5$ ;:NO, CONTINUE  
BR 15$ ;:RESTORE TRAP CATCHER  
  
10$: ADD #4, SP ;:ADJUST STACK  
MOV R2, $BADDR ;:LOAD ADDRESS PRINT OUT  
ERROR 1  
15$: MOV #ERRVEC+2, ERRVEC ;:RESTORE TRAP CATCHER  
CLR ERRVEC+2
```

.SBTTL \*\*RESET, CONTROLLER CLEAR, AND TRI-STATE TESTS

```
*****  
:TEST 2 RESET RK611 AND VERIFY REGISTERS  
:  
:* RESET THE RK611 CONTROLLER AND READ ALL REGISTER OF THE  
:* RK611 REGISTERS EXCEPT THE DATA BUFFER AND VERIFY THAT  
:* THEY ARE CORRECT. REEXAMINE COMMAND AND STATUS REGISTER 1  
:* TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE  
:* COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE  
:* DID NOT SET.  
:  
:* THE SUCCESSFUL EXECUTION OF THIS TEST VERIFIES THAT NO  
:* BIT OF THE TRI-STATE BUS IS STUCK TO ONE.  
:*****
```

```
TST2: SCOPE  
MOV #10., $TIMES ;:DO 10. ITERATIONS  
MOV $BASE, R2 ;:LOAD RK611 BASE REGISTER  
MOV #EM2, EM2N ;:LOAD ERROR MESSAGE FOR PRINT OUT  
RESET ;:CLEAR RK611 WITH UNIBUS INIT  
JSR PC, CHKPAR ;:CHECK FOR MEMORY CHECK ENABLE  
MOV RKCS1(R2), $BDDAT ;:SAVE COMMAND AND STATUS REG.7  
CMP #RDY, $BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 1$ ;:YES, CHECK OTHER REGISTERS  
MOV #RDY, $GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EMJ000, EM2N+2 ;:LOAD ERROR MESSAGE
```



1787	003236	104002				ERROR	2	
1788	003240	016237	000004	001126	1\$:	MOV	RKBA(R2), \$BDDAT	; SAVE BUS ADD REG
1789	003246	001406				BEQ	2\$	; CHECK IF ZERO
1790	003250	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1791	003254	012737	066721	001312		MOV	#EM1002, EM2N+2	; LOAD ERROR MESSAGE
1792	003262	104002				ERROR	2	
1793	003264	016237	000006	001126	2\$:	MOV	RKDA(R2), \$BDDAT	; SAVE DISK ADDRESS ERROR
1794	003272	001406				BEQ	3\$	; CHECK IF ZERO
1795	003274	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1796	003300	012737	066743	001312		MOV	#EM1003, EM2N+2	; LOAD ERROR MESSAGE
1797	003306	104002				ERROR	2	
1798	003310	016237	000016	001126	3\$:	MOV	RKASOF(R2), \$BDDAT	; STORE ATTENTION/OFFSET REG.
1799	003316	001406				BEQ	4\$	; CHECK IF ZERO
1800	003320	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1801	003324	012737	067063	001312		MOV	#EM1007, EM2N+2	; LOAD ERROR MESSAGE
1802	003332	104002				ERROR	2	
1803	003334	016237	000010	001126	4\$:	MOV	RKCS2(R2), \$BDDAT	; STORE COMMAND AND STATUS REG 2
1804	003342	022737	000100	001126		CMP	#IR, \$BDDAT	; CHECK IF CS2 CORRECT
1805	003350	001407				BEQ	5\$	; YES, CONTINUE
1806	003352	012737	000100	001124		MOV	#IR, \$GDDAT	; LOAD EXPECTED CONTENTS
1807	003360	012737	066766	001312		MOV	#EM1004, EM2N+2	; LOAD ERROR MESSAGE
1808	003366	104002				ERROR	2	
1809	003370	016237	000012	001126	5\$:	MOV	RKDS(R2), \$BDDAT	; STORE DRIVE STATUS REG
1810	003376	001406				BEQ	6\$	; CHECK IF ZERO
1811	003400	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1812	003404	012737	067004	001312		MOV	#EM1005, EM2N+2	; LOAD ERROR MESSAGE
1813	003412	104002				ERROR	2	
1814	003414	016237	000014	001126	6\$:	MOV	RKER(2), \$BDDAT	; STORE ERROR STATUS REG
1815	003422	001406				BEQ	7\$	; CHECK IF ZERO
1816	003424	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1817	003430	012737	067037	001312		MOV	#EM1006, EM2N+2	; LOAD ERROR MESSAGE
1818	003436	104002				ERROR	2	
1819	003440	016237	000020	001126	7\$:	MOV	RKDCYL(R2), \$BDDAT	; STORE CYLINDER ADD REG
1820	003446	001406				BEQ	9\$	; CHECK IF EQUAL ZERO
1821	003450	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED CONTENTS
1822	003454	012737	067354	001312		MOV	#EM1016, EM2N+2	; LOAD ERROR MESSAGE
1823	003462	104002				ERROR	2	
1824	003464	016237	000026	001126	9\$:	MOV	RKMR1(R2), \$BDDAT	; STORE MAINTENANCE REG 1
1825	003472	012737	002000	001124		MOV	#MEWD, \$GDDAT	; LOAD EXPECTED MR1
1826	003500	032737	020000	001126		BIT	#ECCW, \$BDDAT	
1827	003506	001403				BEQ	10\$	
1828	003510	052737	020000	001124		BIS	#ECCW, \$GDDAT	
1829	003516	023737	001124	001126	10\$:	CMP	\$GDDAT, \$BDDAT	; CHECK IF MR1 CORRECT
1830	003524	001407				BEQ	11\$	; YES, CONTINUE TEST
1831	003526	012737	022000	001124		MOV	#ECCW!MEWD, \$GDDAT	; LOAD EXPECTED CONTENTS
1832	003534	012737	067154	001312		MOV	#EM1009, EM2N+2	; LOAD ERROR MESSAGE
1833	003542	104002				ERROR	2	
1834	003544	016237	000032	001126	11\$:	MOV	RKECPT(R2), \$BDDAT	; STORE ECC PATTERN REG.
1835	003552	001406				BEQ	12\$	; CHECK IF ZERO
1836	003554	005037	001124			CLR	\$GDDAT	; LOAD EXPECTED RESULTS
1837	003560	012737	067224	001312		MOV	#EM1013, EM2N+2	; LOAD ERROR MESSAGE
1838	003566	104002				ERROR	2	
1839	003570	016237	000030	001126	12\$:	MOV	RKECPS(R2), \$BDDAT	; STORE ECC POSITION REG.
1840	003576	022737	004066	001126		CMP	#4066, \$BDDAT	; CHECK IF ECC POSITION CORRECT
1841	003604	001407				BEQ	13\$	; YES, CONTINUE
1842	003606	012737	004066	001124		MOV	#4066, \$GDDAT	; LOAD EXPECTED RESULTS



1843	003614	012737	067202	001312		MOV	#EM1012,EM2N+2	;LOAD ERROR MESSAGE
1844	003622	104002				ERROR	2	
1845	003624	016237	000000	001126	13\$:	MOV	RKCS1(R2), \$BDDAT	;STORE COMMAND AND STATUS REG 1
1846	003632	022737	000200	001126		CMP	#RDY, \$BDDAT	;CHECK IF CS1 CORRECT
1847	003640	001407				BEQ	14\$	;YES, CONTINUE
1848	003642	012737	000200	001124		MOV	#RDY, \$GDDAT	;LOAD EXPECTED RESULTS
1849	003650	012737	067246	001312		MOV	#EM1014,EM2N+2	;LOAD ERROR MESSAGE
1850	003656	104002				ERROR	2	
1851	003660	016237	000010	001126	14\$:	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG 2.
1852	003666	022737	000100	001126		CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
1853	003674	001407				BEQ	TST3	::GO ON TO NEXT TEXT
1854	003676	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED RESULTS
1855	003704	012737	067311	001312		MOV	#EM1015,EM2N+2	;LOAD ERROR MESSAGE
1856	003712	104002				ERROR	2	
1857								



1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871 003714 000004  
1872 003716 012737 000144 001200  
1873 003724 013702 001270  
1874 003730 012737 063465 001310  
1875 003736 012762 100000 000000  
1876 003744 016237 000000 001126  
1877 003752 022737 000200 001126  
1878 003760 001407  
1879 003762 012737 000200 001124  
1880 003770 012737 066656 001312  
1881 003776 104002  
1882 004000 016237 000004 001126 1\$:  
1883 004006 001406  
1884 004010 005037 001124  
1885 004014 012737 066721 001312  
1886 004022 104002  
1887 004024 016237 000006 001126 2\$:  
1888 004032 001406  
1889 004034 005037 001124  
1890 004040 012737 066743 001312  
1891 004046 104002  
1892 004050 016237 000016 001126 3\$:  
1893 004056 001406  
1894 004060 005037 001124  
1895 004064 012737 067063 001312  
1896 004072 104002  
1897 004074 016237 000010 001126 4\$:  
1898 004102 022737 000100 001126  
1899 004110 001407  
1900 004112 012737 000100 001124  
1901 004120 012737 066766 001312  
1902 004126 104002  
1903 004130 016237 000012 001126 5\$:  
1904 004136 001406  
1905 004140 005037 001124  
1906 004144 012737 067004 001312  
1907 004152 104002  
1908 004154 016237 000014 001126 6\$:  
1909 004162 001406  
1910 004164 005037 001124  
1911 004170 012737 067037 001312  
1912 004176 104002  
1913 004200 016237 000020 001126 7\$:

```
*****  
:TEST 3 CONTROLLER CLEAR AND VERIFY REGISTERS  
:  
: INITIALIZE THE RK611 CONTROLLER WITH A CONTROLLER  
: CLEAR AND READ ALL REGISTER OF THE RK611 REGISTERS  
: EXCEPT THE DATA BUFFER AND VERIFY THAT THEY ARE  
: CORRECT. REEXAMINE COMMAND AND STATUS REGISTER 1  
:  
: TO MAKE SURE CONTROLLER ERROR DID NOT SET. REEXAMINE  
: COMMAND AND STATUS REGISTER 2 TO MAKE SURE DATA LATE  
: DID NOT SET.  
:  
*****  
TST3: SCOPE  
MOV #100, $TIMES ;; DO 100. ITERATIONS  
MOV $BASE, R2 ;LOAD RK611 BASE REGISTER  
MOV #EM3, EM2N ;LOAD ERROR MESSAGE FOR PRINT OUT  
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV RKCS1(R2), $BDDAT ;SAVE COMMAND AND STATUS REG.7  
CMP #RDY, $BDDAT ;CHECK IF CS1 CORRECT  
BEQ 1$ ;YES, CHECK OTHER REGISTERS  
MOV #RDY, $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1000, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKBA(R2), $BDDAT ;SAVE BUS ADD REG  
BEQ 2$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1002, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKDA(R2), $BDDAT ;SAVE DISK ADDRESS ERROR  
BEQ 3$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1003, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKASOF(R2), $BDDAT ;STORE ATTENTION/OFFSET REG.  
BEQ 4$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1007, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG 2  
CMP #IR, $BDDAT ;CHECK IF CS2 CORRECT  
BEQ 5$ ;YES, CONTINUE  
MOV #IR, $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1004, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG  
BEQ 6$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1005, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKER(2), $BDDAT ;STORE ERROR STATUS REG  
BEQ 7$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1006, EM2N+2 ;LOAD ERROR MESSAGE  
ERROR 2  
MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
```



```

1914 004206 001406 BEQ 9$ :CHECK IF EQUAL ZERO
1915 004210 005037 001124 CLR $GDDAT :LOAD EXPECTED CONTENTS
1916 004214 012737 067354 001312 MOV #EM1016,EM2N+2 :LOAD ERROR MESSAGE
1917 004222 104002 ERROR 2
1918 004224 016237 000026 001126 9$: MOV RKM(R2), $BDDAT :STORE MAINTENANCE REG 1
1919 004232 012737 002000 001124 MOV #MEWD, $GDDAT :LOAD EXPECTED MR1
1920 004240 032737 020000 001126 BIT #ECCW, $BDDAT
1921 004246 001403 BEQ 10$
1922 004250 052737 020000 001124 RIS #ECCW, $GDDAT
1923 004256 023737 001124 001126 10$: CMP $GDDAT, $BDDAT :CHECK IF MR1 CORRECT
1924 004264 001407 BEQ 11$ :YES, CONTINUE TEST
1925 004266 012737 022000 001124 MOV #ECCW!MEWD, $GDDAT :LOAD EXPECTED CONTENTS
1926 004274 012737 067154 001312 MOV #EM1009, EM2N+2 :LOAD ERROR MESSAGE
1927 004302 104002 ERROR 2
1928 004304 016237 000032 001126 11$: MOV RKECPT(R2), $BDDAT :STORE ECC PATTERN REG.
1929 004312 001406 BEQ 12$ :CHECK IF ZERO
1930 004314 005037 001124 CLR $GDDAT :LOAD EXPECTED RESULTS
1931 004320 012737 067224 001312 MOV #EM1013, EM2N+2 :LOAD ERROR MESSAGE
1932 004326 104002 ERROR 2
1933 004330 016237 000030 001126 12$: MOV RKECPS(R2), $BDDAT :STORE ECC POSITION REG.
1934 004336 022737 004066 001126 CMP #4066, $BDDAT :CHECK IF ECC POSITION CORRECT
1935 004344 001407 BEQ 13$ :YES, CONTINUE
1936 004346 012737 004066 001124 MOV #4066, $GDDAT :LOAD EXPECTED RESULTS
1937 004354 012737 067202 001312 MOV #EM1012, EM2N+2 :LOAD ERROR MESSAGE
1938 004362 104002 ERROR 2
1939 004364 016237 000000 001126 13$: MOV RKCS1(R2), $BDDAT :STORE COMMAND AND STATUS REG 1
1940 004372 022737 000200 001126 CMP #RDY, $BDDAT :CHECK IF CS1 CORRECT
1941 004400 001407 BEQ 14$ :YES, CONTINUE
1942 004402 012737 000200 001124 MOV #RDY, $GDDAT :LOAD EXPECTED RESULTS
1943 004410 012737 067246 001312 MOV #EM1014, EM2N+2 :LOAD ERROR MESSAGE
1944 004416 104002 ERROR 2
1945 004420 016237 000010 001126 14$: MOV RKCS2(R2), $BDDAT :STORE COMMAND AND STATUS REG 2.
1946 004426 022737 000100 001126 CMP #IR, $BDDAT :CHECK IF CS2 CORRECT
1947 004434 001407 BEQ TST4 :GO ON TO NEXT TEXT
1948 004436 012737 000100 001124 MOV #IR, $GDDAT :LOAD EXPECTED RESULTS
1949 004444 012737 067311 001312 MOV #EM1015, EM2N+2 :LOAD ERROR MESSAGE
1950 004452 104002 ERROR 2
  
```

```

1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962 004454 000004
1963 004456 012737 000012 001200
1964 004464 013702 001270
1965 004470 012737 177777 002010
1966 004476 012737 063547 001320
1967 004504 012762 100000 000000
1968 004512 012762 177777 000004
1969 004520 016237 000004 001126
  
```

```

*****
: *TEST 4 WRITE BUS ADDRESS WITH 177777 (PART 1)
: *
: * THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777
: * AND CHECK IF EQUAL TO 177776 AND THAT NO REGISTER INTERACTION
: * TAKES PLACE. A RESET IS DONE AT THE END OF THE TEST TO
: * MAKE SURE THE BUS ADDRESS CLEARS AND ALL RK611 REGISTERS
: * ARE IN THEIR INITIALIZED STATE.
: *
*****
  
```

```

TST4: SCOPE
MOV #10, $TIMES :DO 10. ITERATIONS
MOV $BASE, R2 :LOAD RK611 BASE
MOV #177777, CONFIG :LOAD VALUE WRITTEN FOR PRINT OUT
MOV #EM4, EM3N :LOAD ERROR MESSAGE
MOV #CCLR, RKCS1(R2) :INITIALIZE RK611 CONTROLLER
MOV #177777, RKBA(R2) :WRITE BUS ADDRESS WITH 177777
MOV RKBA(R2), $BDDAT :STORE BUS ADDRESS REG
  
```



1970	004526	022737	177776	001126		CMP	#177776,\$BDDAT	:CHECK IF BUS ADDRESS CORRECT
1971	004534	001407				BEQ	1\$	:YES, CHECK IF ANY REGISTER MODIFIED
1972	004536	012737	177776	001124		MOV	#177776,\$GDDAT	:LOAD EXPECTED RESULTS
1973	004544	012737	066721	001322		MOV	#EM1002,EM3N+2	:LOAD ERROR MESSAGE
1974	004552	104003				ERROR	3	
1975	004554	016237	000000	001126	1\$:	MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
1976	004562	022737	000200	001126		CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
1977	004570	001407				BEQ	2\$	:YES, CHECK OTHER REGISTERS
1978	004572	012737	000200	001124		MOV	#RDY,\$GDDAT	:LOAD EXPECTED RESULTS
1979	004600	012737	067376	001322		MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
1980	004606	104003				ERROR	3	
1981	004610	016237	000006	001126	2\$:	MOV	RKDA(R2),\$BDDAT	:STORE DISK ADD REG.
1982	004616	001406				BEQ	3\$	:CHECK IF ZERO
1983	004620	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
1984	004624	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
1985	004632	104003				ERROR	3	
1986	004634	016237	000016	001126	3\$:	MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
1987	004642	001406				BEQ	4\$	:CHECK IF ZERO
1988	004644	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
1989	004650	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
1990	004656	104003				ERROR	3	
1991	004660	016237	000010	001126	4\$:	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG 2
1992	004666	022737	000100	001126		CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
1993	004674	001407				BEQ	5\$	:YES, CONTINUE
1994	004676	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
1995	004704	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
1996	004712	104003				ERROR	3	
1997	004714	016237	000012	001126	5\$:	MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG.
1998	004722	001406				BEQ	6\$	:CHECK IF ZERO
1999	004724	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2000	004730	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
2001	004736	104003				ERROR	3	
2002	004740	016237	000014	001126	6\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG.
2003	004746	001406				BEQ	7\$	:CHECK IF ZERO
2004	004750	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2005	004754	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
2006	004762	104003				ERROR	3	
2007	004764	016237	000020	001126	7\$:	MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD. REG..
2008	004772	001406				BEQ	9\$	:CHECK IF EQUAL ZERO
2009	004774	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED RESULTS
2010	005000	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
2011	005006	104003				ERROR	3	
2012	005010	016237	000026	001126	9\$:	MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG. 1
2013	005016	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
2014	005024	032737	020000	001126		BIT	#ECCW,\$BDDAT	
2015	005032	001403				BEQ	10\$	
2016	005034	032737	020000	001124		BIS	#ECCW,\$GDDAT	
2017	005042	032737	001124	001126	10\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
2018	005050	001404				BEQ	11\$	:YES, CONTINUE TEST
2019	005052	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
2020	005060	104003				ERROR	3	
2021	005062	016237	000032	001126	11\$:	MOV	RKECPT(R2),\$BDDAT	:STORE ECC PATTERN REG.
2022	005070	001406				BEQ	12\$	:CHECK IF ZERO
2023	005072	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED RESULTS
2024	005076	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
2025	005104	104003				ERROR	3	



```

2026 005106 016237 000030 001126 12$: MOV RKECP5(R2), $BDDAT ;STORE ECC POSITION REGC
2027 005114 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF ECC POSITION CRRECT
2028 005122 001407 BEQ 13$ ;YES, CLEAR RK611
2029 005124 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED RESULTS
2030 005132 012737 067764 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
2031 005140 104003 ERROR 3
2032 005142 012737 177776 002014 13$: MOV #177776, PREREG ;LOAD PREVIOUS CONTENTS
2033 005150 012737 063603 001330 MOV #EM5, EM4N ;LOAD ERROR MESSAGE
2034 005156 000005 RESET ;CLEAR RK611 WITH UNIBUS INIT
2035 005160 004737 055264 JSR PC, CHKPAR ;CHECK FOR MEMORY CHECK ENABLE
2036 005164 016237 000004 001126 MOV RKBA(R2), $BDDAT ;STORE BUS ADDRESS REG.
2037 005172 001406 BEQ TST5 ;CHECK IF CLEARED
2038 005174 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
2039 005200 012737 066721 001332 MOV #EM1002, EM4N+2 ;LOAD ERROR MESSAGE
2040 005206 104004 ERROR 4
  
```

```

2041
2042 *****
2043 *TEST 5 WRITE BUS ADDRESS WITH 177777 (PART 2)
2044 *
2045 * THIS TEST WILL WRITE THE BUS ADDRESS REGISTER TO 177777.
2046 * A CONTROLLER CLEAR IS DONE. MAKE SURE THE BUS ADDRESS CLEARS.
2047 *
  
```

```

2048 *****
2049 TST5: SCOPE
2050 005210 000004 MOV #100, $TIMES ;DO 100. ITERATIONS
2051 005212 012737 000144 001200 MOV $BASE, R2 ;LOAD RK611 BASE
2052 005220 013702 001270 MOV #CCLR, RKCS1(R2) ;CLEAR KK611 WITH CONTROLLER CLEAR
2053 005224 012762 100000 000000 MOV #177777, RKBA(R2) ;LOAD BUS ADDRESS WITH ALL ONES
2054 005232 012762 177777 000004 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
2055 005240 012762 100000 000000 MOV RKBA(R2), $BDDAT ;STORE BUS ADDRESS
2056 005246 016237 000004 001126 BEQ TST6 ;CHECK IF ZERO
2057 005254 001414 MOV #177776, PREREG ;LOAD PREVIOUS CONTENTS
2058 005256 012737 177776 002014 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2059 005264 005037 001124 MOV #EM6, EM4N ;LOAD ERROR MESSAGE
2060 005270 012737 063662 001330 MOV #EM1002, EM4N+2
2061 005276 012737 066721 001332 ERROR 4
2062
  
```

```

2063 *****
2064 *TEST 6 WRITE WORD COUNT REG. WITH 177777
2065 *
2066 * THIS TEST WILL WRITE THE WORD COUNT REGISTER
2067 * TO 0 AND 177777 AND CHECK IF EQUAL TO 0 AND 177777
2068 * RESPECTIVELY AND THAT NO REGISTER INTERACTION TAKES PLACE.
2069 *
2070 * ISSUE A CONTROLLER CLEAR AND MADE SURE THAT THE WORD
2071 * COUNT REGISTER DOES NOT CHANGE.
2072 *
  
```

```

2073 *****
2074 TST6: SCOPE
2075 005306 000004 MOV #500, $TIMES ;DO 500. ITERATIONS
2076 005310 012737 000764 001200 MOV $BASE, R2 ;LOAD RK611 BASE
2077 005316 013702 001270 CLR CONFIG ;CLEAR CONFIGURATION WORD
2078 005322 005037 002010 MOV #EM7, EM3N ;LOAD ERROR MESSAGE
2079 005326 012737 063745 001320 MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
2080 ; SUBTEST LOOP
2081
  
```







```

2138 005724 023737 001124 001126 12$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
2139 005732 001404 BEQ 13$ ;YES, CONTINUE TEST
2140 005734 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2141 005742 104003 ERROR 3
2142 005744 016237 000032 001126 13$: MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
2143 005752 001406 BEQ 14$ ;CHECK IF ZERO
2144 005754 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
2145 005760 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2146 005766 104003 ERROR 3
2147 005770 016237 000030 001126 14$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
2148 005776 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION CORRECT
2149 006004 001407 BEQ 15$ ;YES,CLEAR RK611
2150 006006 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED RESULTS
2151 006014 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2152 006022 104003 ERROR 3
2153 006024 012762 100000 000000 15$: MOV #CCLR,RKCS1(R2) ;CLEAR RK611
2154 006032 016237 000002 001126 MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG.
2155 006040 023737 002010 001126 CMP CONFIG,$BDDAT ;CHECK IF WORD COUNT NOT CHANGED
2156 ; BY COTROLLER CLEAR
2157 006046 001412 BEQ 20$ ;YES, CONTINUE
2158 006050 013737 002010 001124 MOV CONFIG,$GDDAT ;LOAD EXPECTED DATA
2159 006056 012737 063465 001310 MOV #EM3,EM2N ;LOAD ERROR MESSAGE
2160 006064 012737 067417 001312 MOV #EM1018,EM2N+2
2161 006072 104002 ERROR 2
2162 006074 104415 20$: SCOPE1 ;TEST IF LOOP ON ERROR
2163 006076 005737 002010 TST CONFIG ;CHECK IF FINISHED
2164 006102 001005 BNE TST7 ;:YES, GO TO NEXT TEST
2165 006104 012737 177777 002010 MOV #177777,CONFIG ;USE 177777 FOR SECOND PASS
2166 006112 000137 005342 JMP 1$
  
```

```

*****
*TEST 7 WRITE DISK ADDRESS WITH 177777
*
* THIS TEST WILL WRITE THE DISK ADDRESS REGISTER TO 177777
* AND CHECK IF EQUAL TO 003437 AND THAT NO REGISTER INTERACTION
* TAKES PLACE. A CONTROLLER CLEAR IS DONE AT THE END OF
* THE TEST TO MAKE SURE THE DISK ADDRESS CLEARS.
*****
  
```

```

2177 006116 000004 TST7: SCOPE
2178 006120 012737 000144 001200 MOV #100, $TIMES ;:DO 100. ITERATIONS
2179 006126 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE
2180 006132 012737 177777 002010 MOV #177777,CONFIG ;LOAD CONFIGURAION WORD
2181 006140 012737 064004 001320 MOV #EM8,EM3N ;LOAD ERROR MESSAGE
2182 006146 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
2183 006154 005062 000002 000006 CLR RKWC(R2) ;CLEAR WORD COUNT REG
2184 006160 012762 177777 000006 MOV #177777,RKDA(R2) ;WRITE DISK ADDRESS WITH
2185 ; 177777
2186 006166 016237 000006 001126 MOV RKDA(R2),$BDDAT ;STORE DISK ADDRESS REG.
2187 006174 022737 003437 001126 CMP #3437,$BDDAT ;CHECK IF DISK ADDRESS REG CORRECT
2188 006202 001407 BEQ 1$ ;YES, CHECK IF AY REGISTER MODIFIED
2189 006204 012737 003437 001124 MOV #3437,$GDDAT ;LOAD EXPECTED RESULTS
2190 006212 012737 066743 001322 MOV #EM1003,EM3N+2 ;LOAD ERROR MESSAGE
2191 006220 104003 ERROR 3
2192 006222 016237 000000 001126 1$: MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
2193 006230 022737 000200 001126 CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
  
```



2194	006236	001407				BEQ	2\$		:YES, CONTINUE
2195	006240	012737	000200	001124		MOV	#RDY,\$GDDAT		:LOAD EXPECTED RESULTS
2196	006246	012737	067376	001322		MOV	#EM1017,EM3N+2		:LOAD ERROR MESSAGE
2197	006254	104003				ERROR	3		
2198	006256	016237	000004	001126	2\$:	MOV	RKBA(R2),\$BDDAT		:STORE BUS ADD REG.
2199	006264	001406				BEQ	3\$		:CHECK IF ZERO
2200	006266	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2201	006272	012737	067447	001322		MOV	#EM1019,EM3N+2		:LOAD ERROR MESSAGE
2202	006300	104003				ERROR	3		
2203	006302	016237	000002	001126	3\$:	MOV	RKWC(R2),\$BDDAT		:STORE WORD COUNT REG.
2204	006310	001406				BEQ	4\$		:CHECK IF ZERO
2205	006312	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2206	006316	012737	067417	001322		MOV	#EM1018,EM3N+2		:LOAD ERROR MESSAGE
2207	006324	104003				ERROR	3		
2208	006326	016237	000016	001126	4\$:	MOV	RKASOF(R2),\$BDDAT		:STORE ATTENTION SUMMARY/OFFSET REG.
2209	006334	001406				BEQ	5\$		:CHECK IF ZERO
2210	006336	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2211	006342	012737	067640	001322		MOV	#EM1024,EM3N+2		:LOAD ERROR MESSAGE
2212	006350	104003				ERROR	3		
2213	006352	016237	000010	001126	5\$:	MOV	RKCS2(R2),\$BDDAT		:STORE COMMAND AND STATUS REG.2
2214	006360	022737	000100	001126		CMP	#IR,\$BDDAT		:CHECK IF CS2 CORRECT
2215	006366	001407				BEQ	6\$		:YES,CONTINUE
2216	006370	012737	000100	001124		MOV	#IR,\$GDDAT		:LOAD EXPECTED CONTENTS
2217	006376	012737	067532	001322		MOV	#EM1021,EM3N+2		:LOAD ERROR MESSAGE
2218	006404	104003				ERROR	3		
2219	006406	016237	000012	001126	6\$:	MOV	RKDS(R2),\$BDDAT		:STORE DRIVE STAUUS REG.
2220	006414	001406				BEQ	7\$		:CHECK IF ZERO
2221	006416	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2222	006422	012737	067553	001322		MOV	#EM1022,EM3N+2		:LOAD ERROR MESSAGE
2223	006430	104003				ERROR	3		
2224	006432	016237	000014	001126	7\$:	MOV	RKER(R2),\$BDDAT		:STORE ERRPR REG.
2225	006440	001406				BEQ	8\$		:CHECK IF ZERO
2226	006442	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2227	006446	012737	067611	001322		MOV	#EM1023,EM3N+2		:LOAD ERROR MESSAGE
2228	006454	104003				ERROR	3		
2229	006456	016237	000020	001126	8\$:	MOV	RKDCYL(R2),\$BDDAT		:STORE CYLINDER ADD REG
2230	006464	001406				BEQ	10\$		:CHECK IF EQUAL ZERO
2231	006466	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED RESULTS
2232	006472	012737	067706	001322		MOV	#EM1025,EM3N+2		:LOAD ERROR MESSAGE
2233	006500	104003				ERROR	3		
2234	006502	016237	000026	001126	10\$:	MOV	RKMR1(R2),\$BDDAT		:STORE MAINTENANCE REG 1
2235	006510	012737	002000	001124		MOV	#MEWD,\$GDDAT		:LOAD EXPECTED MR1
2236	006516	032737	020000	001126		BIT	#ECCW,\$BDDAT		
2237	006524	001403				BEQ	11\$		
2238	006526	052737	020000	001124		BIS	#ECCW,\$GDDAT		
2239	006534	023737	001124	001126	11\$:	CMP	\$GDDAT,\$BDDAT		:CHECK IF MR1 CORRECT
2240	006542	001404				BEQ	12\$		:YES,CONTINUE TEST
2241	006544	012737	067733	001322		MOV	#EM1026,EM3N+2		:LOAD ERROR MESSAGE
2242	006552	104003				ERROR	3		
2243	006554	016237	000032	001126	12\$:	MOV	RKECPT(R2),\$BDDAT		:STORE ECC PATTERN REG.
2244	006562	001406				BEQ	13\$		:CHECK IF ZERO
2245	006564	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED RESULTS
2246	006570	012737	070011	001322		MOV	#EM1030,EM3N+2		:LOAD ERROR MESSAGE
2247	006576	104003				ERROR	3		
2248	006600	016237	000030	001126	13\$:	MOV	RKECPS(R2),\$BDDAT		:STORE ECC POSITION REG.
2249	006606	022737	004066	001126		CMP	#4066,\$BDDAT		:CHECK IF ECC POSITION CORRECT



2250	006614	001407				BEQ	14\$		:YES, ISSUE CONTROLLER CLEAR
2251	006616	012737	004066	001124		MOV	#4066,\$GDDAT		:LOAD EXPECTED RESULTS
2252	006624	012737	067764	001322		MOV	#EM1029,EM3N+2		:LOAD ERROR MESSAGE
2253	006632	104003				ERROR	3		
2254	006634	012762	100000	000000	14\$:	MOV	#CCLR,RKCS1(R2)		:CLEAR RK611
2255	006642	016237	000006	001126		MOV	RKDA(R2),\$BDDAT		:STORE DISK ADDRESS REG.
2256	006650	001414				BEQ	TST10		::YES, GO ON TO NEXT TEST
2257	006652	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
2258	006656	012737	063465	001330		MOV	#EM3,EM4N		:LOAD ERROR MESSAGE
2259	006664	012737	066743	001332		MOV	#EM1003,EM4N+2		
2260	006672	012737	003437	002014		MOV	#3437,PREREG		:LOAD PREVIOUS VALUE
2261	006700	104004				ERROR	4		

2262  
2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292

```

*****
SBTTL **REGISTER INTERACTION TESTS
*
* ALL REGISTER INTERACTION TESTS CONSISTS OF WRITING A
* REGISTER AND CHECKING ITS CONTENTS AGAINST EXPECTED
* CONTENTS. THEN ALL OTHER REGISTERS ARE READ EXCEPT
* THE DATA BUFFER TO CHECK WHETHER THEY HAVE CHANGED FROM
* THEIR INITIALIZED CONDITIONS.
*****

```

```

*****
*TEST 10 REGISTER INTERACTION USING BUS ADDRESS (PART 1)
*
* THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
* CLEAR TO HE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
* COUNT REGISTER TO 0.
*
* THE TEST ITSELF WILL CONSIST OF WRITING THE
* BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
* TEST IF BUS ADDRESS IS CORRECT AND THAT NO
* REGISTER INTERACTION TAKES PLACE.
*
* 000000 000010 000200 004000 100000
* 000001 000020 000400 010000
* 000002 000040 001000 020000
* 000004 000100 002000 040000
*****

```

2293	006702	000004				TST10:	SCOPE		
2294	006704	012737	000144	001200		MOV	#100,\$TIMES		::DO 100. ITERATIONS
2295	006712	012701	000021			MOV	#17,R1		:LOAD NUMBER OF PATTERNS
2296	006716	012737	000001	002010		MOV	#000001,CONFIG		:LOAD INITIAL CONFIGURATION
2297	006724	012737	063547	001320		MOV	#EM4,EM3N		:LOAD ERROR MESSAGE
2298	006732	012762	100000	000000		MOV	#CCLR,RKCS1(R2)		:CLEAR RK611 WITH CONTROLLER CLEAR
2299	006740	012737	006746	001110		MOV	#1\$,\$LPERR		:LOAD LOOP ON ERROR LOCATION FOR
2300									: SUBTEST LOOP
2301									
2302	006746				1\$:	CLR	RKWC(R2)		:CLEAR WORD COUNT REG.
2303	006746	005062	000002			MOV	CONFIG,RKBA(R2)		:WRITE RKBA
2304	006752	013762	002010	000004		MOV	RKBA(R2),\$BDDAT		:STORE RKBA
2305	006760	016237	000004	001126					



```

2306 006766 013737 002010 001124 MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
2307 006774 042737 000001 001124 BIC #1,$GDDAT ;INITIALIZE READ ONLY BITS
2308 007002 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKBA CORRECT
2309 007010 001404 BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED
2310 007012 012737 066721 001322 MOV #EM1002,EM3N+2 ;LOAD ERROR MESSAGE
2311 007020 104003 ERROR 3
2312 007022 2$:
2313 007022 016237 000000 001126 MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
2314 007030 022737 000200 001126 CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
2315 007036 001407 BEQ 3$ ;YES, CONTINUE
2316 007040 012737 000100 001124 MOV #IR,$GDDAT ;LOAD EXPECTED RESULTS
2317 007046 012737 067376 001322 MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
2318 007054 104003 ERROR 3
2319 007056 3$:
2320 007056 016237 000002 001126 MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG
2321 007064 001406 BEQ 5$ ;CHECK IF ZERO
2322 007066 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2323 007072 012737 067417 001322 MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
2324 007100 104003 ERROR 3
2325 007102 5$:
2326 007102 016237 000006 001126 MOV RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
2327 007110 001406 BEQ 6$ ;CHECK IF ZERO
2328 007112 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2329 007116 012737 067474 001322 MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
2330 007124 104003 ERROR 3
2331 007126 6$:
2332 007126 016237 000016 001126 MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2333 007134 001406 BEQ 7$ ;CHECK IF ZERO
2334 007136 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2335 007142 012737 067640 001322 MOV #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
2336 007150 104003 ERROR 3
2337 007152 7$:
2338 007152 016237 000010 001126 MOV RKCS2(R2),$BDDAT ;STORE COMMAND AND STATUS REG.2
2339 007160 022737 000100 001126 CMP #IR,$BDDAT ;CHECK IF CS2 CORRECT
2340 007166 001407 BEQ 8$ ;YES,CONTINUE
2341 007170 012737 000100 001124 MOV #IR,$GDDAT ;LOAD EXPECTED CONTENTS
2342 007176 012737 067532 001322 MOV #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
2343 007204 104003 ERROR 3
2344 007206 8$:
2345 007206 016237 000012 001126 MOV RKDS(R2),$BDDAT ;STORE DRIVE STATUS REG
2346 007214 001406 BEQ 9$ ;CHECK IF ZERO
2347 007216 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2348 007222 012737 067553 001322 MOV #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
2349 007230 104003 ERROR 3
2350 007232 016237 000014 001126 9$: MOV RKER(R2),$BDDAT ;STORE ERROR REG
2351 007240 001406 BEQ 10$ ;CHECK IF ZERO
2352 007242 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2353 007246 012737 067611 001322 MOV #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
2354 007254 104003 ERROR 3
2355 007256 10$:
2356 007256 016237 000020 001126 MOV RKDCYL(R2),$BDDAT ;STORE CYLINDER ADD REG
2357 007264 001406 BEQ 12$ ;CHECK IF ZERO
2358 007266 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2359 007272 012737 067706 001322 MOV #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
2360 007300 104003 ERROR 3
2361 007302 12$:

```



```

2362 007302 016237 000026 001126 MOV RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
2363 007310 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MR1
2364 007316 032737 020000 001126 BIT #ECCW, $BDDAT
2365 007324 001403 BEQ 13$
2366 007326 052737 020000 001124 BIS #ECCW, $GDDAT
2367 007334 023737 001124 001126 13$: CMP $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
2368 007342 001404 BEQ 14$ ;YES, ISSUE CONTROLLER CLEAR
2369 007344 012737 067733 001322 MOV #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
2370 007352 104003 ERROR 3
2371 007354 14$:
2372 007354 016237 000032 001126 MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2373 007362 001406 BEQ 15$ ;CHECK IF ZERO
2374 007364 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2375 007370 012737 070011 001322 MOV #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
2376 007376 104003 ERROR 3
2377 007400 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2378 007406 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
2379 007414 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2380 007422 001404 BEQ 18$ ;YES, INITIALIZE RK611
2381 007424 012737 067764 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
2382 007432 104003 ERROR 3
2383 007434 016237 000004 002014 18$: MOV RKBA(R2), PREREG ;GET PREVIOUS CONTENTS
2384 007442 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2385 007450 016237 000004 001126 MOV RKBA(R2), $BDDAT ;GET CURRENT VALUE
2386 007456 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2387 007462 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKBA CORRECT
2388 007470 001407 BEQ 19$ ;YES, CHECK IF FINISHED
2389 007472 012737 063465 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
2390 007500 012737 066721 001332 MOV #EM1002, EM4N+2
2391 007506 104004 ERROR 4
2392 007510 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
2393 007512 000241 CLC ;SHIFT IN ZERO
2394 007514 006137 002010 ROL CONFIG
2395 007520 005301 DEC R1 ;CHECK IF FINISHED
2396 007522 001402 BEQ TST11 ;:YES, GO ON TO NEXT TEST
2397 007524 000137 006746 JMP 1$
2398
2399

```

```

2400 *****
2401 *TEST 11 REGISTER INTERACTION USING BUS ADDRESS (PART 2)
2402 *
2403 * THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2404 * CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
2405 * COUNT REGISTER TO 0.
2406 *
2407 * THE TEST ITSELF WILL CONSIST OF WRITING THE
2408 * BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND
2409 * TEST IF BUS ADDRESS IS CORRECT AND THAT NO
2410 * REGISTER INTERACTION TAKES PLACE.
2411 *
2412 * 177777 177767 177577 173777 077777
2413 * 177776 177757 177377 167777
2414 * 177775 177737 176777 157777
2415 * 177773 177677 175777 137777
2416 *
2417 *****
2417 007530 000004 TST11: SCOPE

```



2418	007532	012737	000144	001200	MOV	#100,\$TIMES	::DO 100. ITERATIONS
2419	007540	012701	000021		MOV	#17,R1	:LOAD NUMBER OF PATTERNS
2420	007544	012737	177776	002010	MOV	#177776,CONFIG	:LOAD INITIAL CONFIGURATION
2421	007552	012737	063547	001320	MOV	#EM4,EM3N	:LOAD ERROR MESSAGE
2422	007560	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	:CLEAR RK611 WITH CONTROLLER CLEAR
2423	007566	012737	007574	001110	MOV	#1\$,\$LPERR	:LOAD LOOP ON ERROR LOCATION FOR
2424							: SUBTEST LOOP
2425							
2426	007574				1\$:		
2427	007574	005062	000002		CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
2428	007600	013762	002010	000004	MOV	CONFIG,RKBA(R2)	:WRITE RKBA
2429	007606	016237	000004	001126	MOV	RKBA(R2),\$BDDAT	:STORE RKBA
2430	007614	013737	002010	001124	MOV	CONFIG,\$GDDAT	:PREPARE EXPECTED RESULTS
2431	007622	042737	000001	001124	BIC	#1,\$GDDAT	:INITIALIZE READ ONLY BITS
2432	007630	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKBA CORRECT
2433	007636	001404			BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
2434	007640	012737	066721	001322	MOV	#EM1002,EM3N+2	:LOAD ERROR MESSAGE
2435	007646	104003			ERROR	3	
2436	007650				2\$:		
2437	007650	016237	000000	001126	MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
2438	007656	022737	000200	001126	CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
2439	007664	001407			BEQ	3\$	:YES, CONTINUE
2440	007666	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
2441	007674	012737	067376	001322	MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
2442	007702	104003			ERROR	3	
2443	007704				3\$:		
2444	007704	016237	000002	001126	MOV	RKWC(R2),\$BDDAT	:STORE WORD COUNT REG
2445	007712	001406			BEQ	5\$	:CHECK IF ZERO
2446	007714	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2447	007720	012737	067417	001322	MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
2448	007726	104003			ERROR	3	
2449	007730				5\$:		
2450	007730	016237	000006	001126	MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
2451	007736	001406			BEQ	6\$	:CHECK IF ZERO
2452	007740	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2453	007744	012737	067474	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
2454	007752	104003			ERROR	3	
2455	007754				6\$:		
2456	007754	016237	000016	001126	MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
2457	007762	001406			BEQ	7\$	:CHECK IF ZERO
2458	007764	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2459	007770	012737	067640	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
2460	007776	104003			ERROR	3	
2461	010000				7\$:		
2462	010000	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
2463	010006	022737	000100	001126	CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
2464	010014	001407			BEQ	8\$	:YES,CONTINUE
2465	010016	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
2466	010024	012737	067532	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
2467	010032	104003			ERROR	3	
2468	010034				8\$:		
2469	010034	016237	000012	001126	MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
2470	010042	001406			BEQ	9\$	:CHECK IF ZERO
2471	010044	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2472	010050	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
2473	010056	104003			ERROR	3	



```

2474 010060 016237 000014 001126 9$: MOV RKER(R2), $BDDAT ;STORE ERROR REG
2475 010066 001406 BEQ 10$ ;CHECK IF ZERO
2476 010070 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2477 010074 012737 067611 001322 MOV #EM1023, EM3N+2 ;LOAD ERROR MESSAGE
2478 010102 104003 ERROR 3
2479 010104 10$:
2480 010104 016237 000020 001126 MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
2481 010112 001406 BEQ 12$ ;CHECK IF ZERO
2482 010114 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2483 010120 012737 067706 001322 MOV #EM1025, EM3N+2 ;LOAD ERROR MESSAGE
2484 010126 104003 ERROR 3
2485 010130 12$:
2486 010130 016237 000026 001126 MOV RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
2487 010136 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MR1
2488 010144 032737 020000 001126 BIT #ECCW, $BDDAT
2489 010152 001403 BEQ 13$
2490 010154 052737 020000 001124 BIS #ECCW, $GDDAT
2491 010162 023737 001124 001126 13$: CMP $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
2492 010170 001404 BEQ 14$ ;YES, ISSUE CONTROLLER CLEAR
2493 010172 012737 067733 001322 MOV #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
2494 010200 104003 ERROR 3
2495 010202 14$:
2496 010202 016237 000032 001126 MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2497 010210 001406 BEQ 15$ ;CHECK IF ZERO
2498 010212 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2499 010216 012737 070011 001322 MOV #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
2500 010224 104003 ERROR 3
2501 010226 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2502 010234 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
2503 010242 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2504 010250 001404 BEQ 18$ ;YES, INITIALIZE RK611
2505 010252 012737 067764 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
2506 010260 104003 ERROR 3
2507 010262 016237 000004 002014 18$: MOV RKBA(R2), PREREG ;GET PREVIOUS CONTENTS
2508 010270 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2509 010276 016237 000004 001126 MOV RKBA(R2), $BDDAT ;GET CURRENT VALUE
2510 010304 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2511 010310 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKBA CORRECT
2512 010316 001407 BEQ 19$ ;YES, CHECK IF FINISHED
2513 010320 012737 063465 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
2514 010326 012737 066721 001332 MOV #EM1002, EM4N+2
2515 010334 104004 ERROR 4
2516 010336 104415 19$: SCOP1 ;CHECK IF LOOP ON ERROR
2517 010340 000261 SEC ;SHIFT IN ONE
2518 010342 006137 002010 ROL CONFIG
2519 010346 005301 DEC R1 ;CHECK IF FINISHED
2520 010350 001402 BEQ TST12 ;:YES, GO ON TO NEXT TEST
2521 010352 000137 007574 JMP 1$
  
```

```

2522
2523 *****
2524 *TEST 12 REGISTER INTERACTION USING BUS ADDRESS (PART 3)
2525 *
2526 *
2527 * THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER
2528 * CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD
2529 * COUNT REGISTER TO 0.
  
```



2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2540  
2541 010356 000004  
2542 010360 012737 000144 001200  
2543 010366 012701 000021  
2544 010372 005037 002010  
2545 010376 012737 063547 001320  
2546 010404 012762 100000 000000  
2547 010412 012737 010420 001110  
2548  
2549  
2550 010420  
2551 010420 005062 000002  
2552 010424 013762 002010 000004  
2553 010432 016237 000004 001126  
2554 010440 013737 002010 001124  
2555 010446 042737 000001 001124  
2556 010454 023737 001124 001126  
2557 010462 001404  
2558 010464 012737 066721 001322  
2559 010472 104003  
2560 010474  
2561 010474 016237 000000 001126  
2562 010502 022737 000200 001126  
2563 010510 001407  
2564 010512 012737 000100 001124  
2565 010520 012737 067376 001322  
2566 010526 104003  
2567 010530  
2568 010530 016237 000002 001126  
2569 010536 001406  
2570 010540 005037 001124  
2571 010544 012737 067417 001322  
2572 010552 104003  
2573 010554  
2574 010554 016237 000006 001126  
2575 010562 001406  
2576 010564 005037 001124  
2577 010570 012737 067474 001322  
2578 010576 104003  
2579 010600  
2580 010600 016237 000016 001126  
2581 010606 001406  
2582 010610 005037 001124  
2583 010614 012737 067640 001322  
2584 010622 104003  
2585 010624

THE TEST ITSELF WILL CONSIST OF WRITING THE  
BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND  
TEST IF BUS ADDRESS IS CORRECT AND THAT NO  
REGISTER INTERACTION TAKES PLACE.

000001 000037 000777 017777 000000  
000003 000077 001777 037777  
000007 000177 003777 077777  
000017 000377 007777 177777

\*\*\*\*\*

TST12: SCOPE  
MOV #100, \$TIMES ;DO 100. ITERATIONS  
MOV #17, R1 ;LOAD NUMBER OF PATTERNS  
CLR CONFIG ;LOAD INITIAL CONFIGURATION  
MOV #EM4, EM3N ;LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$:  
CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MOV CONFIG, RKBA(R2) ;WRITE RKBA  
MOV RKBA(R2), \$BDDAT ;STORE RKBA  
MOV CONFIG, \$GDDAT ;PREPARE EXPECTED RESULTS  
BIC #1, \$GDDAT ;INITIALIZE READ ONLY BITS  
CMP \$GDDAT, \$BDDAT ;CHECK IF RKBA CORRECT  
BEQ 2\$ ;YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1002, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

2\$:  
MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG. 1  
CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT  
BEQ 3\$ ;YES, CONTINUE  
MOV #IR, \$GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

3\$:  
MOV RKWC(R2), \$BDDAT ;STORE WORD COUNT REG  
BEQ 5\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

5\$:  
MOV RKDA(R2), \$BDDAT ;STORE DISK AVERAGE REG  
BEQ 6\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1020, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

6\$:  
MOV RKASOF(R2), \$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.  
BEQ 7\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1024, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

7\$:



2586	010624	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
2587	010632	022737	000100	001126	CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
2588	010640	001407			BEQ	8\$	:YES, CONTINUE
2589	010642	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
2590	010650	012737	067532	001322	MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
2591	010656	104003			ERROR	3	
2592	010660						
2593	010660	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
2594	010666	001406			BEQ	9\$	:CHECK IF ZERO
2595	010670	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2596	010674	012737	067553	001322	MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
2597	010702	104003			ERROR	3	
2598	010704	016237	000014	001126	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
2599	010712	001406			BEQ	10\$	:CHECK IF ZERO
2600	010714	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2601	010720	012737	067611	001322	MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
2602	010726	104003			ERROR	3	
2603	010730						
2604	010730	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
2605	010736	001406			BEQ	12\$	:CHECK IF ZERO
2606	010740	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2607	010744	012737	067706	001322	MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
2608	010752	104003			ERROR	3	
2609	010754						
2610	010754	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
2611	010762	012737	002000	001124	MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
2612	010770	032737	020000	001126	BIT	#ECCW, \$BDDAT	
2613	010776	001403			BEQ	13\$	
2614	011000	052737	020000	001124	BIS	#ECCW, \$GDDAT	
2615	011006	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
2616	011014	001404			BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
2617	011016	012737	067733	001322	MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
2618	011024	104003			ERROR	3	
2619	011026						
2620	011026	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
2621	011034	001406			BEQ	15\$	:CHECK IF ZERO
2622	011036	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2623	011042	012737	070011	001322	MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
2624	011050	104003			ERROR	3	
2625	011052	016237	000030	001126	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
2626	011060	012737	004066	001124	MOV	#4066, \$GDDAT	:USE 4066
2627	011066	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
2628	011074	001404			BEQ	18\$	:YES, INITIALIZE RK611
2629	011076	012737	067764	001322	MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
2630	011104	104003			ERROR	3	
2631	011106	016237	000004	002014	MOV	RKBA(R2), PREREG	:GET PREVIOUS CONTENTS
2632	011114	012762	100000	000000	MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
2633	011122	016237	000004	001126	MOV	RKBA(R2), \$BDDAT	:GET CURRENT VALUE
2634	011130	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
2635	011134	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF RKBA CORRECT
2636	011142	001407			BEQ	19\$	:YES, CHECK IF FINISHED
2637	011144	012737	063465	001330	MOV	#EM3, EM4N	:LOAD ERROR MESSAGE
2638	011152	012737	066721	001332	MOV	#EM1002, EM4N+2	
2639	011160	104004			ERROR	4	
2640	011162	104415			SCOPI		:CHECK IF LOOP ON ERROR
2641	011164	000261			SEC		:SHIFT IN ONE



2642 011166 006137 002010  
2643 011172 005301  
2644 011174 001402  
2645 011176 000137 010420  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664

ROL CONFIG  
DEC R1 ;CHECK IF FINISHED  
BEQ TST13 ;:YES, GO ON TO NEXT TEST  
JMP 1\$

\*\*\*\*\*  
\*TEST 13 REGISTER INTERACTION USING BUS ADDRESS (PART 4)  
\*\*\*\*\*

THIS TEST WILL INITIALIZE ALL REGISTERS WITH A CONTROLLER CLEAR TO THE RK611 CONTROLLER. IT WILL THEN WRITE THE WORD COUNT REGISTER TO 0.

THE TEST ITSELF WILL CONSIST OF WRITING THE BUS ADDRESS REGISTER WITH THE FOLLOWING CONFIGURATIONS AND TEST IF BUS ADDRESS IS CORRECT AND THAT NO REGISTER INTERACTION TAKES PLACE.

100000 174000 177600 177770 000000  
140000 176000 177700 177774  
160000 177000 177740 177776  
170000 177400 177760 177777

2665 011202 000004  
2666 011204 012737 000144 001200  
2667 011212 012701 000021  
2668 011216 005037 002010  
2669 011222 012737 063547 001320  
2670 011230 012762 100000 000000  
2671 011236 012737 011244 001110  
2672  
2673  
2674 011244  
2675 011244 005062 000002  
2676 011250 013762 002010 000004  
2677 011256 016237 000004 001126  
2678 011264 013737 002010 001124  
2679 011272 042737 000001 001124  
2680 011300 023737 001124 001126  
2681 011306 001404  
2682 011310 012737 066721 001322  
2683 011316 104003  
2684 011320  
2685 011320 016237 000000 001126  
2686 011326 022737 000200 001126  
2687 011334 001407  
2688 011336 012737 000100 001124  
2689 011344 012737 067376 001322  
2690 011352 104003  
2691 011354  
2692 011354 016237 000002 001126  
2693 011362 001406  
2694 011364 005037 001124  
2695 011370 012737 067417 001322  
2696 011376 104003  
2697 011400

TST13: SCOPE  
MOV #100, \$TIMES ;:DO 100. ITERATIONS  
MOV #17, R1 ;:LOAD NUMBER OF PATTERNS  
CLR CONFIG ;:LOAD INITIAL CONFIGURATION  
MOV #EM4, EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP  
  
1\$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG, RKBA(R2) ;:WRITE RKBA  
MOV RKBA(R2), \$BDDAT ;:STORE RKBA  
MOV CONFIG, \$GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #1, \$GDDAT ;:INITIALIZE READ ONLY BITS  
CMP \$GDDAT, \$BDDAT ;:CHECK IF RKBA CORRECT  
BEQ 2\$ ;:YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1002, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
2\$:  
MOV RKCS1(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY, \$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3\$ ;:YES, CONTINUE  
MOV #IR, \$GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
3\$:  
MOV RKWC(R2), \$BDDAT ;:STORE WORD COUNT REG  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
5\$:







```
2754 011730 104003          ERROR 3
2755 011732 016237 000004 002014 18$: MOV RKBA(R2),PREREG ;GET PREVIOUS CONTENTS
2756 011740 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
2757 011746 016237 000004 001126 MOV RKBA(R2),$BDDAT ;GET CURRENT VALUE
2758 011754 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2759 011760 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKBA CORRECT
2760 011766 001407 BEQ 19$ ;YES, CHECK IF FINISHED
2761 011770 012737 063465 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
2762 011776 012737 066721 001332 MOV #EM1002,EM4N+2
2763 012004 104004          ERROR 4
2764 012006 104415          19$: SCOP1 ;CHECK IF LOOP ON ERROR
2765 012010 000261 SEC ;SHIFT IN ONE
2766 012012 006037 002010 ROR CONFIG
2767 012016 005301 DEC R1 ;CHECK IF FINISHED
2768 012020 001402 BEQ TST14 ;:YES, GO ON TO NEXT TEST
2769 012022 000137 011244 JMP 1$

*****
*TEST 14 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 1)
*
* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
* BY WRITING THE WORD COUNT TO ZERO.
*
* THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
* AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
* SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
* THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
* REGISTER INTERACTION TAKES PLACE.
*
* 000000 000020 000400 010000
* 000002 000040 001000 020000
* 000004 000100 002000 040000
* 000010 000200 004000
*****
TST14: SCOP1
MOV #100, $TIMES ;:DO 100. ITERATIONS
MOV #17, R1 ;LOAD NUMBER OF PATTERNS
CLR CONFIG ;LOAD INITIAL CONFIGURATION
MOV #000001, CONFIG1
MOV #EM16, EM3N ;LOAD ERROR MESSAGE
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG, RKCS1(R2) ;WRITE RKCS1
MOV RKCS1(R2), $BDDAT ;STORE RKCS1
MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
BIC #DI!SPAR!CTO, $GDDAT ;INITIALIZE READ ONLY BITS
BIS #RDY, $GDDAT
CMP $GDDAT, $BDDAT ;CHECK IF RKCS1 CORRECT
BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1000, EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
```



2810	012160				2\$:		
2811	012160	016237	000004	001126		MOV	RKBA(R2), \$BDDAT ;STORE BUS AND REG
2812	012166	001406				BEQ	4\$ ;CHECK IF ZERO
2813	012170	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2814	012174	012737	067447	001322		MOV	#EM1019, EM3N+2 ;LOAD ERROR MESSAGE
2815	012202	104003				ERROR	3
2816	012204				4\$:		
2817	012204	016237	000002	001126		MOV	RKWC(R2), \$BDDAT ;STORE WORD COUNT REG
2818	012212	001406				BEQ	5\$ ;CHECK IF ZERO
2819	012214	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2820	012220	012737	067417	001322		MOV	#EM1018, EM3N+2 ;LOAD ERROR MESSAGE
2821	012226	104003				ERROR	3
2822	012230				5\$:		
2823	012230	016237	000006	001126		MOV	RKDA(R2), \$BDDAT ;STORE DISK AVERAGE REG
2824	012236	001406				BEQ	6\$ ;CHECK IF ZERO
2825	012240	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2826	012244	012737	067474	001322		MOV	#EM1020, EM3N+2 ;LOAD ERROR MESSAGE
2827	012252	104003				ERROR	3
2828	012254				6\$:		
2829	012254	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
2830	012262	001406				BEQ	7\$ ;CHECK IF ZERO
2831	012264	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2832	012270	012737	067640	001322		MOV	#EM1024, EM3N+2 ;LOAD ERROR MESSAGE
2833	012276	104003				ERROR	3
2834	012300				7\$:		
2835	012300	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT ;STORE COMMAND AND STATUS REG.2
2836	012306	022737	000100	001126		CMP	#IR, \$BDDAT ;CHECK IF CS2 CORRECT
2837	012314	001407				BEQ	8\$ ;YES, CONTINUE
2838	012316	012737	000100	001124		MOV	#IR, \$GDDAT ;LOAD EXPECTED CONTENTS
2839	012324	012737	067532	001322		MOV	#EM1021, EM3N+2 ;LOAD ERROR MESSAGE
2840	012332	104003				ERROR	3
2841	012334				8\$:		
2842	012334	016237	000012	001126		MOV	RKDS(R2), \$BDDAT ;STORE DRIVE STATUS REG
2843	012342	001406				BEQ	9\$ ;CHECK IF ZERO
2844	012344	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2845	012350	012737	067553	001322		MOV	#EM1022, EM3N+2 ;LOAD ERROR MESSAGE
2846	012356	104003				ERROR	3
2847	012360	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT ;STORE ERROR REG
2848	012366	001406				BEQ	10\$ ;CHECK IF ZERO
2849	012370	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2850	012374	012737	067611	001322		MOV	#EM1023, EM3N+2 ;LOAD ERROR MESSAGE
2851	012402	104003				ERROR	3
2852	012404				10\$:		
2853	012404	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT ;STORE CYLINDER ADD REG
2854	012412	001406				BEQ	12\$ ;CHECK IF ZERO
2855	012414	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
2856	012420	012737	067706	001322		MOV	#EM1025, EM3N+2 ;LOAD ERROR MESSAGE
2857	012426	104003				ERROR	3
2858	012430				12\$:		
2859	012430	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT ;STORE MAINTENANCE REG.1
2860	012436	012737	002000	001124		MOV	#MEWD, \$GDDAT ;LOAD EXPECTED MR1
2861	012444	032737	020000	001126		BIT	#ECCW, \$BDDAT
2862	012452	001403				BEQ	13\$
2863	012454	052737	020000	001124		BIS	#ECCW, \$GDDAT
2864	012462	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT ;CHECK IF MR1 CORRECT
2865	012470	001404				BEQ	14\$ ;YES, ISSUE CONTROLLER CLEAR



```

2866 012472 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2867 012500 104003 ERROR 3
2868 012502 14$:
2869 012502 016237 000032 001126 MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
2870 012510 001406 BEQ 15$ ;CHECK IF ZERO
2871 012512 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
2872 012516 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
2873 012524 104003 ERROR 3
2874 012526 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
2875 012534 032737 010000 002010 BIT #CFMT, CONFIG ;CHECK IF IN 18 BIT FORMAT
2876 012542 001404 BEQ 16$ ;NO, USE 4066
2877 012544 012737 005066 001124 MOV #5066, $GDDAT ;USE 5066
2878 012552 000403 BR 17$ ;CHECK IF POSITION CORRECT
2879
2880 012554 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
2881 012562 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
2882 012570 001404 BEQ 18$ ;YES, INITIALIZE RK611
2883 012572 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
2884 012600 104003 ERROR 3
2885 012602 016237 000000 002014 18$: MOV RKCS1(R2), PREREG ;GET PREVIOUS CONTENTS
2886 012610 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
2887 012616 016237 000000 001126 MOV RKCS1(R2), $BDDAT ;GET CURRENT VALUE
2888 012624 012737 000200 001124 MOV #RDY, $GDDAT ;LOAD EXPECTED CONTENTS
2889 012632 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKCS1 CORRECT
2890 012640 001407 BEQ 19$ ;YES, CHECK IF FINISHED
2891 012642 012737 063465 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
2892 012650 012737 066656 001332 MOV #EM1000, EM4N+2
2893 012656 104004 ERROR 4
2894 012660 104415 19$: SCOP1 ;CHECK IF LOOP ON ERROR
2895 012662 000241 LDC ;SHIFT IN ZERO
2896 012664 006137 002012 ROL CONFIG1
2897 012670 013737 002012 002010 MOV CONFIG1, CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
2898 012676 042737 100001 002010 BIC #CCLR!GO, CONFIG ; GO DO NOT SET
2899 012704 005301 DEC R1 ;CHECK IF FINISHED
2900 012706 001402 BEQ TST15 ;:YES, GO ON TO NEXT TEST
2901 012710 000137 012076 JMP 1$
  
```

2902  
 2903  
 2904  
 2905  
 2906  
 2907  
 2908  
 2909  
 2910  
 2911  
 2912  
 2913  
 2914  
 2915  
 2916  
 2917  
 2918  
 2919  
 2920  
 2921 012714 000004

```

*****
*TEST 15 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 2)
*
* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
* BY WRITING THE WORD COUNT TO ZERO.
*
* THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
* AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
* SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
* THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
* REGISTER INTERACTION TAKES PLACE.
*
* 077776 077756 077376 067776
* 077774 077736 076776 057776
* 077772 077676 075776 037776
* 077766 077576 073776
*****
TST15: SCOPE
  
```



```

2922 012716 012737 000144 001200      MOV      #100.,$TIMES      ;;DO 100. ITERATIONS
2923 012724 012701 000021 001200      MOV      #17.,R1          ;;LOAD NUMBER OF PATTERNS
2924 012730 012737 077776 002010      MOV      #077776,CONFIG   ;;LOAD INITIAL CONFIGURATION
2925 012736 012737 177776 002012      MOV      #177776,CONFG1
2926 012744 012737 064403 001320      MOV      #EM16,EM3N       ;;LOAD ERROR MESSAGE
2927 012752 012762 100C00 000000      MOV      #CCLR,RKCS1(R2)  ;;CLEAR RK611 WITH CONTROLLER CLEAR
2928 012760 012737 012766 001110      MOV      #1$, $LPERR      ;;LOAD LOOP ON ERROR LOCATION FOR
2929                                     ;; SUBTEST LOOP
2930
2931 012766                                     1$:
2932 012766 005062 000002 001200      CLR      RKWC(R2)         ;;CLEAR WORD COUNT REG.
2933 012772 013762 002010 000000      MOV      CONFIG,RKCS1(R2) ;;WRITE RKCS1
2934 013000 016237 000000 001126      MOV      RKCS1(R2), $BDDAT ;;STORE RKCS1
2935 013006 013737 002010 001124      MOV      CONFIG,$GDDAT    ;;PREPARE EXPECTED RESULTS
2936 013014 042737 064000 001124      BIC      #DI!SPAR!CTO,$GDDAT ;;INITIALIZE READ ONLY BITS
2937 013022 052737 000200 001124      BIS      #RDY,$GDDAT
2938 013030 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;;CHECK IF RKCS1 CORRECT
2939 013036 001404                                     BEQ      2$               ;;YES,TEST IF ANY OTHER REG MODIFIED
2940 013040 012737 066656 001322      MOV      #EM1000,EM3N+2   ;;LOAD ERROR MESSAGE
2941 013046 104003                                     ERROR    3
2942 013050                                     2$:
2943 013050 016237 000004 001126      MOV      RKBA(R2), $BDDAT ;;STORE BUS AND REG
2944 013056 001406                                     BEQ      4$               ;;CHECK IF ZERO
2945 013060 005037 001124                                     CLR      $GDDAT           ;;LOAD EXPECTED CONTENTS
2946 013064 012737 067447 001322      MOV      #EM1019,EM3N+2   ;;LOAD ERROR MESSAGE
2947 013072 104003                                     ERROR    3
2948 013074                                     4$:
2949 013074 016237 000002 001126      MOV      RKWC(R2), $BDDAT ;;STORE WORD COUNT REG
2950 013102 001406                                     BEQ      5$               ;;CHECK IF ZERO
2951 013104 005037 001124                                     CLR      $GDDAT           ;;LOAD EXPECTED CONTENTS
2952 013110 012737 067417 001322      MOV      #EM1018,EM3N+2   ;;LOAD ERROR MESSAGE
2953 013116 104003                                     ERROR    3
2954 013120                                     5$:
2955 013120 016237 000006 001126      MOV      RKDA(R2), $BDDAT ;;STORE DISK AVERAGE REG
2956 013126 001406                                     BEQ      6$               ;;CHECK IF ZERO
2957 013130 005037 001124                                     CLR      $GDDAT           ;;LOAD EXPECTED CONTENTS
2958 013134 012737 067474 001322      MOV      #EM1020,EM3N+2   ;;LOAD ERROR MESSAGE
2959 013142 104003                                     ERROR    3
2960 013144                                     6$:
2961 013144 016237 000016 001126      MOV      RKASOF(R2), $BDDAT ;;STORE ATTENTION SUMMARY/OFFSET REG.
2962 013152 001406                                     BEQ      7$               ;;CHECK IF ZERO
2963 013154 005037 001124                                     CLR      $GDDAT           ;;LOAD EXPECTED CONTENTS
2964 013160 012737 067640 001322      MOV      #EM1024,EM3N+2   ;;LOAD ERROR MESSAGE
2965 013166 104003                                     ERROR    3
2966 013170                                     7$:
2967 013170 016237 000010 001126      MOV      RKCS2(R2), $BDDAT ;;STORE COMMAND AND STATUS REG.2
2968 013176 022737 000100 001126      CMP      #IR,$BDDAT       ;;CHECK IF CS2 CORRECT
2969 013204 001407                                     BEQ      8$               ;;YES,CONTINUE
2970 013206 012737 00C100 001124      MOV      #IR,$GDDAT       ;;LOAD EXPECTED CONTENTS
2971 013214 012737 067532 001322      MOV      #EM1021,EM3N+2   ;;LOAD ERROR MESSAGE
2972 013222 104003                                     ERROR    3
2973 013224                                     8$:
2974 013224 016237 000012 001126      MOV      RKDS(R2), $BDDAT ;;STORE DRIVE STATUS REG
2975 013232 001406                                     BEQ      9$               ;;CHECK IF ZERO
2976 013234 005037 001124                                     CLR      $GDDAT           ;;LOAD EXPECTED CONTENTS
2977 013240 012737 067553 001322      MOV      #EM1022,EM3N+2   ;;LOAD ERROR MESSAGE
  
```



```
2978 013246 104003          ERROR 3
2979 013250 016237 000014 001126 9$: MOV  RKER(R2), $BDDAT ;STORE ERROR REG
2980 013256 001406          BEQ  10$           ;CHECK IF ZERO
2981 013260 005037 001124  CLR  $GDDAT       ;LOAD EXPECTED CONTENTS
2982 013264 012737 067611 001322  MOV  #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
2983 013272 104003          ERROR 3
2984 013274          10$:
2985 013274 016237 000020 001126  MOV  RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
2986 013302 001406          BEQ  12$           ;CHECK IF ZERO
2987 013304 005037 001124  CLR  $GDDAT       ;LOAD EXPECTED CONTENTS
2988 013310 012737 067706 001322  MOV  #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
2989 013316 104003          ERROR 3
2990 013320          12$:
2991 013320 016237 000026 001126  MOV  RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG.1
2992 013326 012737 002000 001124  MOV  #MEWD, $GDDAT   ;LOAD EXPECTED MR1
2993 013334 032737 020000 001126  BIT  #ECCW, $BDDAT
2994 013342 001403          BEQ  13$
2995 013344 052737 020000 001124  BIS  #ECCW, $GDDAT
2996 013352 023737 001124 001126 13$: CMP  $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
2997 013360 001404          BEQ  14$           ;YES, ISSUE CONTROLLER CLEAR
2998 013362 012737 067733 001322  MOV  #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
2999 013370 104003          ERROR 3
3000 013372          14$:
3001 013372 016237 000032 001126  MOV  RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
3002 013400 001406          BEQ  15$           ;CHECK IF ZERO
3003 013402 005037 001124  CLR  $GDDAT       ;LOAD EXPECTED CONTENTS
3004 013406 012737 070011 001322  MOV  #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3005 013414 104003          ERROR 3
3006 013416 016237 000030 001126 15$: MOV  RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
3007 013424 032737 010000 002010  BIT  #CFMT, CONFIG  ;CHECK IF IN 18 BIT FORMAT
3008 013432 001404          BEQ  16$           ;NO, USE 4066
3009 013434 012737 005066 001124  MOV  #5066, $GDDAT  ;USE 5066
3010 013442 000403          BR   17$           ;CHECK IF POSITION CORRECT
3011
3012 013444 012737 004066 001124 16$: MOV  #4066, $GDDAT  ;USE 4066
3013 013452 023737 001124 001126 17$: CMP  $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
3014 013460 001404          BEQ  18$           ;YES, INITIALIZE RK611
3015 013462 012737 067764 001322  MOV  #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3016 013470 104003          ERROR 3
3017 013472 016237 000000 002014 18$: MOV  RKCS1(R2), PREREG ;GET PREVIOUS CONTENTS
3018 013500 012762 100000 000000  MOV  #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
3019 013506 016237 000000 001126  MOV  RKCS1(R2), $BDDAT ;GET CURRENT VALUE
3020 013514 012737 000200 001124  MOV  #RDY, $GDDAT   ;LOAD EXPECTED CONTENTS
3021 013522 023737 001124 001126  CMP  $GDDAT, $BDDAT ;CHECK IF RKCS1 CORRECT
3022 013530 001407          BEQ  19$           ;YES, CHECK IF FINISHED
3023 013532 012737 063465 001330  MOV  #EM3, EM4N     ;LOAD ERROR MESSAGE
3024 013540 012737 066656 001332  MOV  #EM1000,EM4N+2
3025 013546 104004          ERROR 4
3026 013550 104415          19$: SCOP1           ;CHECK IF LOOP ON ERROR
3027 013552 000261          SEC             ;SHIFT IN ONE
3028 013554 006137 002012  ROL  CONFIG1
3029 013560 013737 002012 002010  MOV  CONFIG1, CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
3030 013566 042737 100001 002010  BIC  #CCLR!GO, CONFIG ; GO DO NOT SET
3031 013574 005301          DEC  R1           ;CHECK IF FINISHED
3032 013576 001402          BEQ  TST16       ;:YES, GO ON TO NEXT TEST
3033 013600 000137 012766          JMP  1$
```



3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089

013604	000004			
013606	012737	000144	001200	
013614	012701	000021		
013620	005037	002010		
013624	005037	002012		
013630	012737	064403	001320	
013636	012762	100000	000000	
013644	012737	013652	001110	
013652				
013652	005062	000002		
013656	013762	002010	000000	
013664	016237	000000	001126	
013672	013737	002010	001124	
013700	042737	064000	001124	
013706	052737	000200	001124	
013714	023737	001124	001126	
013722	001404			
013724	012737	066656	001322	
013732	104003			
013734				
013734	016237	000004	001126	
013742	001406			
013744	005037	001124		
013750	012737	067447	001322	
013756	104003			
013760				
013760	016237	000002	001126	
013766	001406			
013770	005037	001124		
013774	012737	067417	001322	
014002	104003			
014004				
014004	016237	000006	001126	
014012	001406			
014014	005037	001124		

\*\*\*\*\*  
\*TEST 16 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 3)  
\*\*\*\*\*

THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND BY WRITING THE WORD COUNT TO ZERO.

THE TEST ITSELF WILL CONSIST OF WRITING COMMAND AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO REGISTER INTERACTION TAKES PLACE.

000002	000076	001776	037776
000006	000176	003776	077776
000016	000376	007776	000000
000036	000776	017776	

\*\*\*\*\*  
TST16: SCOPE  
\*\*\*\*\*

MOV	#100, \$TIMES	::DO 100. ITERATIONS
MOV	#17, R1	:LOAD NUMBER OF PATTERNS
CLR	CONFIG	:LOAD INITIAL CONFIGURATION
CLR	CONF1	
MOV	#EM16, EM3N	:LOAD ERROR MESSAGE
MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 WITH CONTROLLER CLEAR
MOV	#1\$, \$LPERR	:LOAD LOOP ON ERROR LOCATION FOR : SUBTEST LOOP

1\$:

CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
MOV	CONFIG, RKCS1(R2)	:WRITE RKCS1
MOV	RKCS1(R2), \$BDDAT	:STORE RKCS1
MOV	CONFIG, \$GDDAT	:PREPARE EXPECTED RESULTS
BIC	#DI!SPAR!CTO, \$GDDAT	:INITIALIZE READ ONLY BITS
BIS	#RDY, \$GDDAT	
CMP	\$GDDAT, \$BDDAT	:CHECK IF RKCS1 CORRECT
BEQ	2\$	:YES, TEST IF ANY OTHER REG MODIFIED
MOV	#EM1000, EM3N+2	:LOAD ERROR MESSAGE
ERROR	3	

2\$:

MOV	RKBA(R2), \$BDDAT	:STORE BUS AND REG
BEQ	4\$	:CHECK IF ZERO
CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
ERROR	3	

4\$:

MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG
BEQ	5\$	:CHECK IF ZERO
CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
ERROR	3	

5\$:

MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
BEQ	6\$	:CHECK IF ZERO
CLR	\$GDDAT	:LOAD EXPECTED CONTENTS



3090	014020	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
3091	014026	104003				ERROR	3	
3092	014030				6\$:			
3093	014030	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
3094	014036	001406				BEQ	7\$	:CHECK IF ZERO
3095	014040	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3096	014044	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
3097	014052	104003				ERROR	3	
3098	014054				7\$:			
3099	014054	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
3100	014062	022737	000100	001126		CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
3101	014070	001407				BEQ	8\$	:YES,CONTINUE
3102	014072	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
3103	014100	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
3104	014106	104003				ERROR	3	
3105	014110				8\$:			
3106	014110	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
3107	014116	001406				BEQ	9\$	:CHECK IF ZERO
3108	014120	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3109	014124	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
3110	014132	104003				ERROR	3	
3111	014134	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
3112	014142	001406				BEQ	10\$	:CHECK IF ZERO
3113	014144	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3114	014150	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
3115	014156	104003				ERROR	3	
3116	014160				10\$:			
3117	014160	016237	000020	001126		MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
3118	014166	001406				BEQ	12\$	:CHECK IF ZERO
3119	014170	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3120	014174	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
3121	014202	104003				ERROR	3	
3122	014204				12\$:			
3123	014204	016237	000026	001126		MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG.1
3124	014212	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
3125	014220	032737	020000	001126		BIT	#ECCW,\$BDDAT	
3126	014226	001403				BEQ	13\$	
3127	014230	052737	020000	001124		BIS	#ECCW,\$GDDAT	
3128	014236	023737	001124	001126	13\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
3129	014244	001404				BEQ	14\$	:YES,ISSUE CONTROLLER CLEAR
3130	014246	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
3131	014254	104003				ERROR	3	
3132	014256				14\$:			
3133	014256	016237	000032	001126		MOV	RKECPT(R2),\$BDDAT	:STORE ECC PATTERN REG.
3134	014264	001406				BEQ	15\$	:CHECK IF ZERO
3135	014266	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3136	014272	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
3137	014300	104003				ERROR	3	
3138	014302	016237	000030	001126	15\$:	MOV	RKECPS(R2),\$BDDAT	:STORE ECC POSITION REG.
3139	014310	032737	010000	002010		BIT	#CFMT,CONFIG	:CHECK IF IN 18 BIT FORMAT
3140	014316	001404				BEQ	16\$	:NO, USE 4066
3141	014320	012737	005066	001124		MOV	#5066,\$GDDAT	:USE 5066
3142	014326	000403				BR	17\$	:CHECK IF POSITION CORRECT
3143								
3144	014330	012737	004066	001124	16\$:	MOV	#4066,\$GDDAT	:USE 4066
3145	014336	023737	001124	001126	17\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF ECC POSITION CORRECT



3146 014344 001404  
 3147 014346 012737 067764 001322  
 3148 014354 104003  
 3149 014356 016237 000000 002014  
 3150 014364 012762 100000 000000  
 3151 014372 016237 000C00 001126  
 3152 014400 012737 000200 001124  
 3153 014406 023737 001124 001126  
 3154 014414 001407  
 3155 014416 012737 063465 001330  
 3156 014424 012737 066656 001332  
 3157 014432 104004  
 3158 014434 104415  
 3159 014436 000261  
 3160 014440 006137 002012  
 3161 014444 013737 002012 002010  
 3162 014452 042737 100001 002010  
 3163 014460 005301  
 3164 014462 001402  
 3165 014464 000137 013652

```

BEQ 18$ ;YES, INITIALIZE RK611
MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3
MOV RKCS1(R2),PREREG ;GET PREVIOUS CONTENTS
MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
MOV RKCS1(R2),$BDDAT ;GET CURRENT VALUE
MOV #RDY,$GDDAT ;LOAD EXPECTED CONTENTS
CMP $GDDAT,$BDDAT ;CHECK IF RKCS1 CORRECT
BEQ 19$ ;YES, CHECK IF FINISHED
MOV #EM3,EM4N ;LOAD ERROR MESSAGE
MOV #EM1000,EM4N+2
ERROR 4
19$: SCOP1 ;CHECK IF LOOP ON ERROR
SEC ;SHIFT IN ONE
ROL CONFG1
MOV CONFG1,CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
BIC #CCLR!GO,CONFIG ; GO DO NOT SET
DEC R1 ;CHECK IF FINISHED
BEQ TST17 ;:YES, GO ON TO NEXT TEST
JMP 1$
  
```

```

*****
*TEST 17 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4)
*
* THIS TEST WILL INITIALIZE BY ISSUING A CONTROLLER CLEAR AND
* BY WRITING THE WORD COUNT TO ZERO.
*
* THE TEST ITSELF WILL CONSIST OF WRITING COMMAND
* AND STATUS REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS
* SUCH THAT GO AND CONTROLLER CLEAR ARE RESET. IT WILL
* THEN CHECK FOR CORRECT REGISTER LOADING AND THAT NO
* REGISTER INTERACTION TAKES PLACE.
*
* 000000 074000 077600 077770
* 040000 076000 077700 077774
* 060000 077000 077740 077776
* 070000 077400 077760
*****
  
```

3185 014470 000004  
 3186 014472 012737 000144 001200  
 3187 014500 012701 000021  
 3188 014504 005037 002010  
 3189 014510 005037 002012  
 3190 014514 012737 064403 001320  
 3191 014522 012762 100000 000000  
 3192 014530 012737 014536 001110  
 3193  
 3194  
 3195 014536  
 3196 014536 005062 000002  
 3197 014542 013762 002010 000000  
 3198 014550 016237 000000 001126  
 3199 014556 013737 002010 001124  
 3200 014564 042737 064000 001124  
 3201 014572 052737 000200 001124

```

*****
TST17: SCOPE
MOV #100,$TIMES ;:DO 100. ITERATIONS
MOV #17,R1 ;LOAD NUMBER OF PATTERNS
CLR CONFIG ;LOAD INITIAL CONFIGURATION
CLR CONFG1
MOV #EM16,EM3N ;LOAD ERROR MESSAGE
MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP
1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG,RKCS1(R2) ;WRITE RKCS1
MOV RKCS1(R2),$BDDAT ;STORE RKCS1
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
BIC #DI!SPAR!CTO,$GDDAT ;INITIALIZE READ ONLY BITS
BIS #RDY,$GDDAT
  
```



K 5  
PAGE 63

CZR6AD0 RK611 DSKLS CTRL PRT1 MACY11 30(1046) 14-SEP-81 15:04 REGISTER INTERACTION USING COMMAND STATUS REG.1 (PART 4) SEQ 0062  
CZR6AD.P11 14-SEP-81 13:43 T17

3202	014600	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:CHECK IF RKCS1 CORRECT
3203	014606	001404				BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
3204	014610	012737	066656	001322		MOV	#EM1000,EM3N+2	:LOAD ERROR MESSAGE
3205	014616	104003				ERROR	3	
3206	014620				2\$:			
3207	014620	016237	000004	001126		MOV	RKBA(R2),\$BDDAT	:STORE BUS AND REG
3208	014626	001406				BEQ	4\$	:CHECK IF ZERO
3209	014630	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3210	014634	012737	067447	001322		MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
3211	014642	104003				ERROR	3	
3212	014644				4\$:			
3213	014644	016237	000002	001126		MOV	RKWC(R2),\$BDDAT	:STORE WORD COUNT REG
3214	014652	001406				BEQ	5\$	:CHECK IF ZERO
3215	014654	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3216	014660	012737	067417	001322		MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
3217	014666	104003				ERROR	3	
3218	014670				5\$:			
3219	014670	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
3220	014676	001406				BEQ	6\$	:CHECK IF ZERO
3221	014700	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3222	014704	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
3223	014712	104003				ERROR	3	
3224	014714				6\$:			
3225	014714	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
3226	014722	001406				BEQ	7\$	:CHECK IF ZERO
3227	014724	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3228	014730	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
3229	014736	104003				ERROR	3	
3230	014740				7\$:			
3231	014740	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
3232	014746	022737	090100	001126		CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
3233	014754	001407				BEQ	8\$	:YES,CONTINUE
3234	014756	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
3235	014764	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
3236	014772	104003				ERROR	3	
3237	014774				8\$:			
3238	014774	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
3239	015002	001406				BEQ	9\$	:CHECK IF ZERO
3240	015004	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3241	015010	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
3242	015016	104003				ERROR	3	
3243	015020	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
3244	015026	001406				BEQ	10\$	:CHECK IF ZERO
3245	015030	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3246	015034	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
3247	015042	104003				ERROR	3	
3248	015044				10\$:			
3249	015044	016237	000020	001126		MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
3250	015052	001406				BEQ	12\$	:CHECK IF ZERO
3251	015054	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3252	015060	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
3253	015066	104003				ERROR	3	
3254	015070				12\$:			
3255	015070	016237	000026	001126		MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG.1
3256	015076	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
3257	015104	032737	020000	001126		BIT	#ECCW,\$BDDAT	



```

3258 015112 001403 BEQ 13$
3259 015114 052737 020000 001124 BIS #ECCW,$GDDAT
3260 015122 023737 001124 001126 13$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
3261 015130 001404 BEQ 14$ ;YES,ISSUE CONTROLLER CLEAR
3262 015132 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
3263 015140 104003 ERROR 3
3264 015142 14$:
3265 015142 016237 000032 001126 MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3266 015150 001406 BEQ 15$ ;CHECK IF ZERO
3267 015152 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3268 015156 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3269 015164 104003 ERROR 3
3270 015166 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3271 015174 032737 010000 002010 BIT #CFMT,CONFIG ;CHECK IF IN 18 BIT FORMAT
3272 015202 001404 BEQ 16$ ;NO, USE 4066
3273 015204 012737 005066 001124 MOV #5066,$GDDAT ;USE 5066
3274 015212 000403 BR 17$ ;CHECK IF POSITION CORRECT
3275
3276 015214 012737 004066 001124 16$: MOV #4066,$GDDAT ;USE 4066
3277 015222 023737 001124 001126 17$: CMP $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
3278 015230 001404 BEQ 18$ ;YES,INITIALIZE RK611
3279 015232 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3280 015240 104003 ERROR 3
3281 015242 016237 000000 002014 18$: MOV RKCS1(R2),PREREG ;GET PREVIOUS CONTENTS
3282 015250 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3283 015256 016237 000000 001126 MOV RKCS1(R2),$BDDAT ;GET CURRENT VALUE
3284 015264 012737 000200 001124 MOV #RDY,$GDDAT ;LOAD EXPECTED CONTENTS
3285 015272 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKCS1 CORRECT
3286 015300 001407 BEQ 19$ ;YES, CHECK IF FINISHED
3287 015302 012737 063465 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
3288 015310 012737 066656 001332 MOV #EM1000,EM4N+2
3289 015316 104004 ERROR 4
3290 015320 104415 19$: SCOP1 ;CHECK IF LOOP ON ERROR
3291 015322 000261 SEC ;SHIFT IN ONE
3292 015324 006037 002012 ROR CONFIG1
3293 015330 013737 002012 002010 MOV CONFIG1,CONFIG ;MAKE SURE CONTROLLER CLEAR AMP
3294 015336 042737 100001 002010 BIC #CCLR!GO,CONFIG ; GO DO NOT SET
3295 015344 005301 DEC R1 ;CHECK IF FINISHED
3296 015346 001402 BEQ TST20 ;:YES, GO ON TO NEXT TEST
3297 015350 000137 014536 JMP 1$
3298
3299

```

```

*****
*TEST 20 REGISTER INTERACTION USING SPARE REG
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH 0.
*
* WRITE THE SPARE REGISTER WITH 177777 AND MAKE SURE
* NO INTERACTION TAKES PLACE.
*****

```

```

3300
3301
3302
3303
3304
3305
3306
3307
3308
3309 015354 000004 TST20: SCOPE
3310 015356 012737 000764 001200 MOV #500, $TIMES ;:DO 500. ITERATIONS
3311 015364 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE ADDRESS
3312 015370 012737 177777 002010 MOV #177777,CONFIG ;LOAD CONFIGURATION WORD
3313 015376 012737 064041 001320 MOV #EM9,EM3N ;LOAD ERROR MESSAGE

```



3314	015404	012762	100000	000000		MOV	#CCLR,RKCS1(R2)	;CLEAR RK611 WITH CONTROLLER CLEAR
3315	015412	005062	000002			CLR	RKWC(R2)	;CLEAR WORD COUNT REG.
3316	015416	012762	177777	000022		MOV	#177777,RKSPAR(R2)	;WRITE RKSPAR WITH 177777
3317	015424	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.1
3318	015432	022737	000200	001126		CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
3319	015440	001407				BEQ	1\$	;YES,CONTINUE
3320	015442	012737	000200	001124		MOV	#RDY,\$GDDAT	;LOAD EXPECTED RESULTS
3321	015450	012737	067376	001322		MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
3322	015456	104003				ERROR	3	
3323	015460	016237	000004	001126	1\$:	MOV	RKBA(R2), \$BDDAT	;STORE BUS ADD REG.
3324	015466	001406				BEQ	2\$	;CHECK IF ZERO
3325	015470	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3326	015474	012737	067447	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
3327	015502	104003				ERROR	3	
3328	015504	016237	000002	001126	2\$:	MOV	RKWC(R2), \$BDDAT	;STORE WORK COUNT REG.
3329	015512	001406				BEQ	3\$	;CHECK IF ZERO
3330	015514	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3331	015520	012737	067417	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
3332	015526	104003				ERROR	3	
3333	015530	016237	000006	001126	3\$:	MOV	RKDA(R2), \$BDDAT	;STORE DISK ADD REG
3334	015536	001406				BEQ	4\$	;CHECK IF ZERO
3335	015540	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3336	015544	012737	067474	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
3337	015552	104003				ERROR	3	
3338	015554	016237	000016	001126	4\$:	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG
3339	015562	001406				BEQ	5\$	;CHECK IF ZERO
3340	015564	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3341	015570	012737	067640	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
3342	015576	104003				ERROR	3	
3343	015600	016237	000010	001126	5\$:	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 2
3344	015606	022737	000100	001126		CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
3345	015614	001407				BEQ	6\$	;YES,CONTINUE
3346	015616	012737	000100	001124		MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
3347	015624	012737	067532	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
3348	015632	104003				ERROR	3	
3349	015634	016237	000012	001126	6\$:	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG.
3350	015642	001406				BEQ	7\$	;CHECK IF ZERO
3351	015644	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3352	015650	012737	067553	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
3353	015656	104003				ERROR	3	
3354	015660	016237	000014	001126	7\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
3355	015666	001406				BEQ	8\$	;CHECK IF ZERO
3356	015670	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
3357	015674	012737	067611	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
3358	015702	104003				ERROR	3	
3359	015704	016237	000020	001126	8\$:	MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
3360	015712	001406				BEQ	10\$	;CHECK IF ZERO
3361	015714	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED RESULTS
3362	015720	012737	067706	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
3363	015726	104003				ERROR	3	
3364	015730	016237	000026	001126	10\$:	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG 1
3365	015736	012737	002000	001124		MOV	#MEWD,\$GDDAT	;LOAD EXPECTED CONTENTS
3366	015744	032737	020000	001126		BIT	#ECCW,\$BDDAT	
3367	015752	001403				BEQ	11\$	
3368	015754	052737	020000	001124		BIS	#ECCW,\$GDDAT	
3369	015762	023737	001124	001126	11\$:	CMP	\$GDDAT,\$BDDAT	;CHECK IF MR1 CORRECT



```

3370 015770 001404 BEQ 12$ :YES,CONTINUE TEST
3371 015772 012737 067733 001322 MOV #EM1026,EM3N+2 :LOAD ERROR MESSAGE
3372 016000 104003 ERROR 3
3373 016002 016237 000032 001126 12$: MOV RKECPT(R2), $BDDAT :STORE ECC PATTERN REG.
3374 016010 001406 BEQ 13$ :CHECK IF ZERO
3375 016012 005037 001124 CLR $GDDAT :LOAD EXPECTED CONTENTS
3376 016016 012737 070011 001322 MOV #EM1030,EM3N+2 :LOAD ERROR MESSAGE
3377 016024 104003 ERROR 3
3378 016026 016237 000030 001126 13$: MOV RKECPS(R2), $BDDAT :STORE ECC POSITION REG:
3379 016034 022737 004066 001126 CMP #4066, $BDDAT :CHECK IF ECC POSITION CORRECT
3380 016042 001407 BEQ TST21 :YES,GO TO NEXT TEST
3381 016044 012737 004066 001124 MOV #4066, $GDDAT :LOAD EXPECTED CONTENTS
3382 016052 012737 067764 001322 MOV #EM1029,EM3N+2 :LOAD ERROR MESSAGE
3383 016060 104003 ERROR 3
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
  
```

```

*****
*TEST 21 REGISTER INTERACTION USING WORD COUNT (PART 1)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
* AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
* PLACE.
*
* 000000 000010 000200 004000 100000
* 000001 000020 000400 010000
* 000002 000040 001000 020000
* 000004 000100 002000 040000
  
```

```

3399 016062 000004 TST21: SCOPE
3400 016064 012737 000764 001200 MOV #500, $TIMES :DO 500. ITERATIONS
3401 016072 012701 000021 MOV #17, R1 :LOAD NUMBER OF PATTERNS
3402 016076 012737 000001 002010 MOV #000001, CONFIG :LOAD INITIAL CONFIGURATION
3403 016104 012737 063745 001320 MOV #EM7, EM3N :LOAD ERROR MESSAGE
3404 016112 012762 100000 000000 MOV #CCLR, RKCS1(R2) :CLEAR RK611 WITH CONTROLLER CLEAR
3405 016120 012737 016126 001110 MOV #1$, $LPERR :LOAD LOOP ON ERROR LOCATION FOR
3406 : SUBTEST LOOP
3407
3408 016126 1$: MOV CONFIG, RKWC(R2) :WRITE RKWC
3409 016126 013762 002010 000002 MOV RKWC(R2), $BDDAT :STORE RKWC
3410 016134 016237 000002 001126 MOV CONFIG, $GDDAT :PREPARE EXPECTED RESULTS
3411 016142 013737 002010 001124 CMP $GDDAT, $BDDAT :CHECK IF RKWC CORRECT
3412 016150 023737 001124 001126 BEQ 2$ :YES,TEST IF ANY OTHER REG MODIFIED
3413 016156 001404 MOV #EM1001, EM3N+2 :LOAD ERROR MESSAGE
3414 016160 012737 066674 001322 ERROR 3
3415 016166 104003
3416 016170 2$: MOV RKCS1(R2), $BDDAT :STORE COMMAND AND STATUS REG. 1
3417 016170 016237 000000 001126 CMP #RDY, $BDDAT :CHECK IF CS1 CORRECT
3418 016176 022737 000200 001126 BEQ 3$ :YES, CONTINUE
3419 016204 001407 MOV #IR, $GDDAT :LOAD EXPECTED RESULTS
3420 016206 012737 000100 001124 MOV #EM1017, EM3N+2 :LOAD ERROR MESSAGE
3421 016214 012737 067376 001322 ERROR 3
3422 016222 104003
3423 016224 3$: MOV RKBA(R2), $BDDAT :STORE BUS AND REG
3424 016224 016237 000004 001126 BEQ 4$ :CHECK IF ZERO
3425 016232 001406
  
```



3426	016234	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3427	016240	012737	067447	001322	MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
3428	016246	104003			ERROR	3	
3429	016250				4\$:		
3430	016250	016237	000006	001126	MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
3431	016256	001406			BEQ	6\$	:CHECK IF ZERO
3432	016260	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3433	016264	012737	067474	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
3434	016272	104003			ERROR	3	
3435	016274				6\$:		
3436	016274	016237	000016	001126	MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
3437	016302	001406			BEQ	7\$	:CHECK IF ZERO
3438	016304	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3439	016310	012737	067640	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
3440	016316	104003			ERROR	3	
3441	016320				7\$:		
3442	016320	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
3443	016326	022737	000100	001126	CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
3444	016334	001407			BEQ	8\$	:YES,CONTINUE
3445	016336	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
3446	016344	012737	067532	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
3447	016352	104003			ERROR	3	
3448	016354				8\$:		
3449	016354	016237	000012	001126	MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
3450	016362	001406			BEQ	9\$	:CHECK IF ZERO
3451	016364	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3452	016370	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
3453	016376	104003			ERROR	3	
3454	016400	016237	000014	001126	9\$:		
3455	016406	001406			MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
3456	016410	005037	001124		BEQ	10\$	:CHECK IF ZERO
3457	016414	012737	067611	001322	CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3458	016422	104003			MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
3459	016424				ERROR	3	
3460	016424	016237	000020	001126	10\$:		
3461	016432	001406			MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
3462	016434	005037	001124		BEQ	12\$	:CHECK IF ZERO
3463	016440	012737	067706	001322	CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3464	016446	104003			MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
3465	016450				ERROR	3	
3466	016450	016237	000026	001126	12\$:		
3467	016456	012737	002000	001124	MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG.1
3468	016464	032737	020000	001126	MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
3469	016472	001403			BIT	#ECCW,\$BDDAT	
3470	016474	052737	020000	001124	BEQ	13\$	
3471	016502	023737	001124	001126	BIS	#ECCW,\$GDDAT	
3472	016510	001404			13\$:		
3473	016512	012737	067733	001322	CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
3474	016520	104003			BEQ	14\$	:YES,ISSUE CONTROLLER CLEAR
3475	016522				MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
3476	016522	016237	000032	001126	ERROR	3	
3477	016530	001406			14\$:		
3478	016532	005037	001124		MOV	RKECPT(R2),\$BDDAT	:STORE ECC PATTERN REG.
3479	016536	012737	070011	001322	BEQ	15\$	:CHECK IF ZERO
3480	016544	104003			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3481	016546	016237	000030	001126	MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
					ERROR	3	
					15\$:		
					MOV	RKECPS(R2),\$BDDAT	:STORE ECC POSITION REG.



```

3482 016554 012737 004066 001124 16$: MOV #4066,$GDDAT :USE 4066
3483 016562 023737 001124 001126 17$: CMP $GDDAT,$BDDAT :CHECK IF ECC POSITION CORRECT
3484 016570 001404 BEQ 18$ :YES,INITIALIZE RK611
3485 016572 012737 067764 001322 MOV #EM1029,EM3N+2 :LOAD ERROR MESSAGE
3486 016600 104003 ERROR 3
3487 016602 016237 000002 002014 18$: MOV RKWC(R2),PREREG :GET PREVIOUS CONTENTS
3488 016610 012762 100000 000000 MOV #CCLR,RKCS1(R2) :CLEAR RK611 CONTROLLER
3489 016616 016237 000002 001126 MOV RKWC(R2),$BDDAT :GET CURRENT VALUE
3490 016624 023737 002010 001126 CMP CONFIG,$BDDAT :CHECK IF WORD COUNT NOT CHANGE
3491 : BY CONTROLLER CLEAR
3492 016632 001412 BEQ 19$ :YES, CHECK IF FINISHED
3493 016634 013737 002010 001124 MOV CONFIG,$GDDAT :LOAD EXPECTED DATA
3494 016642 012737 063465 001310 MOV #EM3,EM2N :LOAD ERROR MESSAGE
3495 016650 012737 067417 001312 MOV #EM1018,EM2N+2
3496 016656 104002 ERROR 2
3497 016660 104415 19$: SCOP1 :CHECK IF LOOP ON ERROR
3498 016662 000241 CLC :SHIFT IN ZERO
3499 016664 006137 002010 ROL CONFIG
3500 016670 005301 DEC R1 :CHECK IF FINISHED
3501 016672 001402 BEQ TST22 :;YES, GO ON TO NEXT TEST
3502 016674 000137 016126 JMP 1$
  
```

\*\*\*\*\*  
 :\*TEST 22 REGISTER INTERACTION USING WORD COUNT (PART 2)  
 :\*

ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.  
 WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS  
 AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE  
 PLACE.

```

177777 177767 177577 173777 077777
177776 177757 177377 167777
177775 177737 176777 157777
177773 177677 175777 137777
  
```

\*\*\*\*\*

```

3518 016700 000004 TST22: SCOPE
3519 016702 012737 000764 001200 MOV #500, $TIMES :;DO 500. ITERATIONS
3520 016710 012701 000021 MOV #17, R1 :LOAD NUMBER OF PATTERNS
3521 016714 012737 177776 002010 MOV #177776,CONFIG :LOAD INITIAL CONFIGURATION
3522 016722 012737 063745 001320 MOV #EM7,EM3N :LOAD ERROR MESSAGE
3523 016730 012762 100000 000000 MOV #CCLR,RKCS1(R2) :CLEAR RK611 WITH CONTROLLER CLEAR
3524 016736 012737 016744 001110 MOV #1,$LPERR :LOAD LOOP ON ERROR LOCATION FOR
3525 : SUBTEST LOOP
3526
3527 016744 1$: MOV CONFIG,RKWC(R2) :WRITE RKWC
3528 016744 013762 002010 000002 MOV RKWC(R2),$BDDAT :STORE RKWC
3529 016752 016237 000002 001126 MOV CONFIG,$GDDAT :PREPARE EXPECTED RESULTS
3530 016760 013737 002010 001124 CMP $GDDAT,$BDDAT :CHECK IF RKWC CORRECT
3531 016766 023737 001124 001126 BEQ 2$ :YES,TEST IF ANY OTHER REG MODIFIED
3532 016774 001404 MOV #EM1001,EM3N+2 :LOAD ERROR MESSAGE
3533 016776 012737 066674 001322 ERROR 3
3534 017004 104003
3535 017006 2$: MOV RKCS1(R2),$BDDAT :STORE COMMAND AND STATUS REG. 1
3536 017006 016237 000000 001126 CMP #RDY,$BDDAT :CHECK IF CS1 CORRECT
3537 017014 022737 000200 001126
  
```







```

3594 017340      14$:
3595 017340 016237 000032 001126  MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
3596 017346 001406      BEQ 15$ ;CHECK IF ZERO
3597 017350 005037 001124      CLR $GDDAT ;LOAD EXPECTED CONTENTS
3598 017354 012737 070011 001322  MOV #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
3599 017362 104003      ERROR 3
3600 017364 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
3601 017372 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
3602 017400 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
3603 017406 001404      BEQ 18$ ;YES, INITIALIZE RK611
3604 017410 012737 067764 001322  MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
3605 017416 104003      ERROR 3
3606 017420 016237 000002 002014 18$: MOV RKWC(R2), PREREG ;GET PREVIOUS CONTENTS
3607 017426 012762 100000 000000  MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
3608 017434 016237 000002 001126  MOV RKWC(R2), $BDDAT ;GET CURRENT VALUE
3609 017442 023737 002010 001126  CMP CONFIG, $BDDAT ;CHECK IF WORD COUNT NOT CHANGE
3610                                     ; BY CONTROLLER CLEAR
3611 017450 001412      BEQ 19$ ;YES, CHECK IF FINISHED
3612 017452 013737 002010 001124  MOV CONFIG, $GDDAT ;LOAD EXPECTED DATA
3613 017460 012737 063465 001310  MOV #EM3, EM2N ;LOAD ERROR MESSAGE
3614 017466 012737 067417 001312  MOV #EM1018, EM2N+2
3615 017474 104002      ERROR 2
3616 017476 104415      19$: SCOP1 ;CHECK IF LOOP ON ERROR
3617 017500 000261      SEC ;SHIFT IN ONE
3618 017502 006137 002010  ROL CONFIG
3619 017506 005301      DEC R1 ;CHECK IF FINISHED
3620 017510 001402      BEQ TST23 ;: YES, GO ON TO NEXT TEST
3621 017512 000137 016744  JMP 1$
  
```

```

*****
: *TEST 23 REGISTER INTERACTION USING WORD COUNT (PART 3)
: *
: * ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
: * WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
: * AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
: * PLACE.
: *
: * 000001 000037 000777 017777 000000
: * 000003 000077 001777 037777
: * 000007 000177 003777 077777
: * 000017 000377 007777 177777
: *
  
```

```

*****
3636 TST23: SCOPE
3637 017516 000004      MOV #500, $TIMES ;: DO 500. ITERATIONS
3638 017520 012737 000764 001200  MOV #17, R1 ;LOAD NUMBER OF PATTERNS
3639 017526 012701 000021      CLR CONFIG ;LOAD INITIAL CONFIGURATION
3640 017532 005037 002010  MOV #EM7, EM3N ;LOAD ERROR MESSAGE
3641 017536 012737 063745 001320  MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
3642 017544 012762 100000 000000  MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
3643 017552 012737 017560 001110  ; SUBTEST LOOP
3644
3645
3646 017560      1$:
3647 017560 013762 002010 000002  MOV CONFIG, RKWC(R2) ;WRITE RKWC
3648 017566 016237 000002 001126  MOV RKWC(R2), $BDDAT ;STORE RKWC
3649 017574 013737 002010 001124  MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
  
```



3650	017602	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:CHECK IF RKWC CORRECT
3651	017610	001404				BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
3652	017612	012737	066674	001322		MOV	#EM1001,EM3N+2	:LOAD ERROR MESSAGE
3653	017620	104003				ERROR	3	
3654	017622				2\$:			
3655	017622	016237	000000	001126		MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
3656	017630	022737	000200	001126		CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
3657	017636	001407				BEQ	3\$	:YES, CONTINUE
3658	017640	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
3659	017646	012737	067376	001322		MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
3660	017654	104003				ERROR	3	
3661	017656				3\$:			
3662	017656	016237	000004	001126		MOV	RKBA(R2),\$BDDAT	:STORE BUS AND REG
3663	017664	001406				BEQ	4\$	:CHECK IF ZERO
3664	017666	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3665	017672	012737	067447	001322		MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
3666	017700	104003				ERROR	3	
3667	017702				4\$:			
3668	017702	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
3669	017710	001406				BEQ	6\$	:CHECK IF ZERO
3670	017712	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3671	017716	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
3672	017724	104003				ERROR	3	
3673	017726				6\$:			
3674	017726	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
3675	017734	001406				BEQ	7\$	:CHECK IF ZERO
3676	017736	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3677	017742	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
3678	017750	104003				ERROR	3	
3679	017752				7\$:			
3680	017752	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
3681	017760	022737	000100	001126		CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
3682	017766	001407				BEQ	8\$	:YES,CONTINUE
3683	017770	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
3684	017776	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
3685	020004	104003				ERROR	3	
3686	020006				8\$:			
3687	020006	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
3688	020014	001406				BEQ	9\$	:CHECK IF ZERO
3689	020016	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3690	020022	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
3691	020030	104003				ERROR	3	
3692	020032	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
3693	020040	001406				BEQ	10\$	:CHECK IF ZERO
3694	020042	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3695	020046	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
3696	020054	104003				ERROR	3	
3697	020056				10\$:			
3698	020056	016237	000020	001126		MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
3699	020064	001406				BEQ	12\$	:CHECK IF ZERO
3700	020066	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3701	020072	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
3702	020100	104003				ERROR	3	
3703	020102				12\$:			
3704	020102	016237	000026	001126		MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG.1
3705	020110	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1



```

3706 020116 032737 020000 001126 BIT #ECCW,$BDDAT
3707 020124 001403 BEQ 13$
3708 020126 052737 020000 001124 BIS #ECCW,$GDDAT
3709 020134 023737 001124 001126 13$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
3710 020142 001404 BEQ 14$ ;YES,ISSUE CONTROLLER CLEAR
3711 020144 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
3712 020152 104003 ERROR 3
3713 020154 14$:
3714 020154 016237 000032 001126 MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3715 020162 001406 BEQ 15$ ;CHECK IF ZERO
3716 020164 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
3717 020170 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3718 020176 104003 ERROR 3
3719 020200 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3720 020206 012737 004066 001124 16$: MOV #4066,$GDDAT ;USE 4066
3721 020214 023737 001124 001126 17$: CMP $GDDAT,$BDDAT ;CHECK IF ECC POSITION CORRECT
3722 020222 001404 BEQ 18$ ;YES,INITIALIZE RK611
3723 020224 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3724 020232 104003 ERROR 3
3725 020234 016237 000002 002014 18$: MOV RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
3726 020242 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3727 020250 016237 000002 001126 MOV RKWC(R2),$BDDAT ;GET CURRENT VALUE
3728 020256 023737 002010 001126 CMP CONFIG,$BDDAT ;CHECK IF WORD COUNT NOT CHANGE
3729 ; BY CONTROLLER CLEAR
3730 ;YES, CHECK IF FINISHED
3731 020264 001412 BEQ 19$ ;LOAD EXPECTED DATA
3732 020266 013737 002010 001124 MOV CONFIG,$GDDAT ;LOAD ERROR MESSAGE
3733 020274 012737 063465 001310 MOV #EM3,EM2N
3734 020302 012737 067417 001312 MOV #EM1018,EM2N+2
3735 020310 104002 ERROR 2
3736 020312 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
3737 020314 000261 SEC ;SHIFT IN ONE
3738 020316 006137 002010 ROL CONFIG
3739 020322 005301 DEC R1 ;CHECK IF FINISHED
3740 020324 001402 BEQ TST24 ;:YES, GO ON TO NEXT TEST
3741 020326 000137 017560 JMP 1$
    
```

```

*****
:TEST 24 REGISTER INTERACTION USING WORD COUNT (PART 4)
:
:ISSUE A CONTROLLER CLEAR TO INITIALIZE ALL RK611 REGISTERS.
:WRITE THE WORD COUNT REGISTER WITH THE FOLLOWING CONFIGURATIONS
:AND CHECK IF CORRECT AND NO REGISTER INTERACTION TAKE
:PLACE.
:
:100000 174000 177600 177770 000000
:140000 176000 177700 177774
:160000 177000 177740 177776
:170000 177400 177760 177777
:
*****
    
```

```

3755 TST24: SCOPE
3756 020332 000004 MOV #500, $TIMES ;:DO 500. ITERATIONS
3757 020334 012737 000764 001200 MOV #17, R1 ;LOAD NUMBER OF PATTERNS
3758 020342 012701 000021 CLR CONFIG ;LOAD INITIAL CONFIGURATION
3759 020346 005037 002010 MOV #EM7,EM3N ;LOAD ERROR MESSAGE
3760 020352 012737 063745 001320 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
3761 020360 012762 100000 000000
    
```



3762	020366	012737	020374	001110	MOV	#1\$, \$LPERR	:LOAD LOOP ON ERROR LOCATION FOR : SUBTEST LOOP
3763							
3764							
3765	020374				1\$:		
3766	020374	013762	002010	000002	MOV	CONFIG, RKWC(R2)	:WRITE RKWC
3767	020402	016237	000C02	001126	MOV	RKWC(R2), \$BDDAT	:STORE RKWC
3768	020410	013737	002010	001124	MOV	CONFIG, \$GDDAT	:PREPARE EXPECTED RESULTS
3769	020416	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF RKWC CORRECT
3770	020424	001404			BEQ	2\$	:YES, TEST IF ANY OTHER REG MODIFIED
3771	020426	012737	066674	001322	MOV	#EM1001, EM3N+2	:LOAD ERROR MESSAGE
3772	020434	104003			ERROR	3	
3773	020436				2\$:		
3774	020436	016237	000000	001126	MOV	RKCS1(R2), \$BDDAT	:STORE COMMAND AND STATUS REG. 1
3775	020444	022737	000200	001126	CMP	#RDY, \$BDDAT	:CHECK IF CS1 CORRECT
3776	020452	001407			BEQ	3\$	:YES, CONTINUE
3777	020454	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED RESULTS
3778	020462	012737	067376	001322	MOV	#EM1017, EM3N+2	:LOAD ERROR MESSAGE
3779	020470	104003			ERROR	3	
3780	020472				3\$:		
3781	020472	016237	000004	001126	MOV	RKBA(R2), \$BDDAT	:STORE BUS AND REG
3782	020500	001406			BEQ	4\$	:CHECK IF ZERO
3783	020502	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3784	020506	012737	067447	001322	MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
3785	020514	104003			ERROR	3	
3786	020516				4\$:		
3787	020516	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
3788	020524	001406			BEQ	6\$	:CHECK IF ZERO
3789	020526	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3790	020532	012737	067474	001322	MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
3791	020540	104003			ERROR	3	
3792	020542				6\$:		
3793	020542	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
3794	020550	001406			BEQ	7\$	:CHECK IF ZERO
3795	020552	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3796	020556	012737	067640	001322	MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
3797	020564	104003			ERROR	3	
3798	020566				7\$:		
3799	020566	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
3800	020574	022737	000100	001126	CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
3801	020602	001407			BEQ	8\$	:YES, CONTINUE
3802	020604	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
3803	020612	012737	067532	001322	MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
3804	020620	104003			ERROR	3	
3805	020622				8\$:		
3806	020622	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
3807	020630	001406			BEQ	9\$	:CHECK IF ZERO
3808	020632	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3809	020636	012737	067553	001322	MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
3810	020644	104003			ERROR	3	
3811	020646	016237	000014	001126	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
3812	020654	001406			BEQ	10\$	:CHECK IF ZERO
3813	020656	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
3814	020662	012737	067611	001322	MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
3815	020670	104003			ERROR	3	
3816	020672				10\$:		
3817	020672	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG



```

3818 020700 001406          BEQ      12$          ;CHECK IF ZERO
3819 020702 005037 001124    CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
3820 020706 012737 067706 001322  MOV     #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
3821 020714 104003          ERROR    3
3822 020716          12$:
3823 020716 016237 000026 001126  MOV     RMR1(R2),$BDDAT ;STORE MAINTENANCE REG.1
3824 020724 012737 002000 001124  MOV     #MEWD,$GDDAT    ;LOAD EXPECTED MR1
3825 020732 032737 020000 001126  BIT     #ECCW,$BDDAT
3826 020740 001403          REQ     13$
3827 020742 052737 020000 001124  BIS     #ECCW,$GDDAT
3828 020750 023737 001124 001126 13$:  CMP     $GDDAT,$BDDAT  ;CHECK IF MR1 CORRECT
3829 020756 001404          BEQ     14$          ;YES,ISSUE CONTROLLER CLEAR
3830 020760 012737 067733 001322  MOV     #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
3831 020766 104003          ERROR    3
3832 020770          14$:
3833 020770 016237 000032 001126  MOV     RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
3834 020776 001406          BEQ     15$          ;CHECK IF ZERO
3835 021000 005037 001124    CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
3836 021004 012737 070011 001322  MOV     #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
3837 021012 104003          ERROR    3
3838 021014 016237 000030 001126 15$:  MOV     RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
3839 021022 012737 004066 001124 16$:  MOV     #4066,$GDDAT    ;USE 4066
3840 021030 023737 001124 001126 17$:  CMP     $GDDAT,$BDDAT  ;CHECK IF ECC POSITION CORRECT
3841 021036 001404          BEQ     18$          ;YES,INITIALIZE RK611
3842 021040 012737 067764 001322  MOV     #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
3843 021046 104003          ERROR    3
3844 021050 016237 000002 002014 18$:  MOV     RKWC(R2),PREREG ;GET PREVIOUS CONTENTS
3845 021056 012762 100000 000000  MOV     #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
3846 021064 016237 000002 001126  MOV     RKWC(R2),$BDDAT ;GET CURRENT VALUE
3847 021072 023737 002010 001126  CMP     CONFIG,$BDDAT  ;CHECK IF WORD COUNT NOT CHANGE
3848          ; BY CONTROLLER CLEAR
3849 021100 001412          BEQ     19$          ;YES, CHECK IF FINISHED
3850 021102 013737 002010 001124  MOV     CONFIG,$GDDAT  ;LOAD EXPECTED DATA
3851 021110 012737 063465 001310  MOV     #EM3,EM2N      ;LOAD ERROR MESSAGE
3852 021116 012737 067417 001312  MOV     #EM1018,EM2N+2
3853 021124 104002          ERROR    2
3854 021126 104415          19$:  SCOP1          ;CHECK IF LOOP ON ERROR
3855 021130 000261          SEC          ;SHIFT IN ONE
3856 021132 006037 002010  ROR     CONFIG
3857 021136 005301          DEC     R1          ;CHECK IF FINISHED
3858 021140 001402          BEQ     TST25       ;:YES, GO ON TO NEXT TEST
3859 021142 000137 020374  JMP     1$

```

```

3860
3861 *****
3862 *TEST 25 REGISTER INTERACTION USING DISK ADDRESS (PART 1)
3863 *
3864 * ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
3865 * WRITE THE WORD COUNT REGISTER WITH 0.
3866 *
3867 * WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
3868 * THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
3869 * SURE NO INTERACTION TAKES PLACE.
3870 *
3871 * 000000 000010 000200 004000 100000
3872 * 000001 000020 000400 010000
3873 * 000002 000040 001000 020000

```



```
3874
3875
3876
3877 021146 000004
3878 021150 012737 000764 001200
3879 021156 012701 000021
3880 021162 012737 000001 002010
3881 021170 012737 064004 001320
3882 021176 012762 100000 000000
3883 021204 012737 021212 001110
3884
3885
3886 021212
3887 021212 005062 000002
3888 021216 013762 002010 000006
3889 021224 016237 000006 001126
3890 021232 013737 002010 001124
3891 021240 042737 174340 001124
3892 021246 023737 001124 001126
3893 021254 001404
3894 021256 012737 066743 001322
3895 021264 104003
3896 021266
3897 021266 016237 000000 001126
3898 021274 022737 000200 001126
3899 021302 001407
3900 021304 012737 000100 001124
3901 021312 012737 067376 001322
3902 021320 104003
3903 021322
3904 021322 016237 000004 001126
3905 021330 001406
3906 021332 005037 001124
3907 021336 012737 067447 001322
3908 021344 104003
3909 021346
3910 021346 016237 000002 001126
3911 021354 001406
3912 021356 005037 001124
3913 021362 012737 067417 001322
3914 021370 104003
3915 021372
3916 021372 016237 000016 001126
3917 021400 001406
3918 021402 005037 001124
3919 021406 012737 067640 001322
3920 021414 104003
3921 021416
3922 021416 016237 000010 001126
3923 021424 022737 000100 001126
3924 021432 001407
3925 021434 012737 000100 001124
3926 021442 012737 067532 001322
3927 021450 104003
3928 021452
3929 021452 016237 000012 001126

: * 000004 000100 002000 040000
: *
: * *****
TST25: SCOPE
MOV #500.,$TIMES ;:DO 500. ITERATIONS
MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS
MOV #000001,CONFIG ;:LOAD INITIAL CONFIGURATION
MOV #EM8,EM3N ;:LOAD ERROR MESSAGE
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$, $LPERR ;:LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

1$:
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.
MOV CONFIG,RKDA(R2) ;:WRITE RKDA
MOV RKDA(R2), $BDDAT ;:STORE RKDA
MOV CONFIG,$GDDAT ;:PREPARE EXPECTED RESULTS
BIC #174340,$GDDAT ;:INITIALIZE READ ONLY BITS
CMP $GDDAT,$BDDAT ;:CHECK IF RKDA CORRECT
BEQ 2$ ;:YES, TEST IF ANY OTHER REG MODIFIED
MOV #EM1003,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

2$:
MOV RKCS1(R2), $BDDAT ;:STORE COMMAND AND STATUS REG. 1
CMP #RDY,$BDDAT ;:CHECK IF CS1 CORRECT
BEQ 3$ ;:YES, CONTINUE
MOV #IR,$GDDAT ;:LOAD EXPECTED RESULTS
MOV #EM1017,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

3$:
MOV RKBA(R2), $BDDAT ;:STORE BUS AND REG
BEQ 4$ ;:CHECK IF ZERO
CLR $GDDAT ;:LOAD EXPECTED CONTENTS
MOV #EM1019,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

4$:
MOV RKWC(R2), $BDDAT ;:STORE WORD COUNT REG
BEQ 5$ ;:CHECK IF ZERO
CLR $GDDAT ;:LOAD EXPECTED CONTENTS
MOV #EM1018,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

5$:
MOV RKASOF(R2), $BDDAT ;:STORE ATTENTION SUMMARY/OFFSET REG.
BEQ 7$ ;:CHECK IF ZERO
CLR $GDDAT ;:LOAD EXPECTED CONTENTS
MOV #EM1024,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

7$:
MOV RKCS2(R2), $BDDAT ;:STORE COMMAND AND STATUS REG.2
CMP #IR,$BDDAT ;:CHECK IF CS2 CORRECT
BEQ 8$ ;:YES, CONTINUE
MOV #IR,$GDDAT ;:LOAD EXPECTED CONTENTS
MOV #EM1021,EM3N+2 ;:LOAD ERROR MESSAGE
ERROR 3

8$:
MOV RKDS(R2), $BDDAT ;:STORE DRIVE STATUS REG
```



3930	021460	001406				BEQ	9\$		:CHECK IF ZERO
3931	021462	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
3932	021466	012737	067553	001322		MOV	#EM1022,EM3N+2		:LOAD ERROR MESSAGE
3933	021474	104003				ERROR	3		
3934	021476	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT		:STORE ERROR REG
3935	021504	001406				BEQ	10\$		:CHECK IF ZERO
3936	021506	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
3937	021512	012737	067611	001322		MOV	#EM1023,EM3N+2		:LOAD ERROR MESSAGE
3938	021520	104003				ERROR	3		
3939	021522				10\$:				
3940	021522	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT		:STORE CYLINDER ADD REG
3941	021530	001406				BEQ	12\$		:CHECK IF ZERO
3942	021532	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
3943	021536	012737	067706	001322		MOV	#EM1025,EM3N+2		:LOAD ERROR MESSAGE
3944	021544	104003				ERROR	3		
3945	021546				12\$:				
3946	021546	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT		:STORE MAINTENANCE REG.1
3947	021554	012737	002000	001124		MOV	#MEWD, \$GDDAT		:LOAD EXPECTED MR1
3948	021562	032737	020000	001126		BIT	#ECCW, \$BDDAT		
3949	021570	001403				BEQ	13\$		
3950	021572	052737	020000	001124		BIS	#ECCW, \$GDDAT		
3951	021600	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT		:CHECK IF MR1 CORRECT
3952	021606	001404				BEQ	14\$		:YES, ISSUE CONTROLLER CLEAR
3953	021610	012737	067733	001322		MOV	#EM1026,EM3N+2		:LOAD ERROR MESSAGE
3954	021616	104003				ERROR	3		
3955	021620				14\$:				
3956	021620	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT		:STORE ECC PATTERN REG.
3957	021626	001406				BEQ	15\$		:CHECK IF ZERO
3958	021630	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
3959	021634	012737	070011	001322		MOV	#EM1030,EM3N+2		:LOAD ERROR MESSAGE
3960	021642	104003				ERROR	3		
3961	021644	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT		:STORE ECC POSITION REG.
3962	021652	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT		:USE 4066
3963	021660	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT		:CHECK IF ECC POSITION CORRECT
3964	021666	001404				BEQ	18\$		:YES, INITIALIZE RK611
3965	021670	012737	067764	001322		MOV	#EM1029,EM3N+2		:LOAD ERROR MESSAGE
3966	021676	104003				ERROR	3		
3967	021700	016237	000006	002014	18\$:	MOV	RKDA(R2), PREREG		:GET PREVIOUS CONTENTS
3968	021706	012762	100000	000000		MOV	#CCLR, RKCS1(R2)		:CLEAR RK611 CONTROLLER
3969	021714	016237	000006	001126		MOV	RKDA(R2), \$BDDAT		:GET CURRENT VALUE
3970	021722	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
3971	021726	023737	001124	001126		CMP	\$GDDAT, \$BDDAT		:CHECK IF RKDA CORRECT
3972	021734	001407				BEQ	19\$		:YES, CHECK IF FINISHED
3973	021736	012737	063465	001330		MOV	#EM3, EM4N		:LOAD ERROR MESSAGE
3974	021744	012737	066743	001332		MOV	#EM1003,EM4N+2		
3975	021752	104004				ERROR	4		
3976	021754	104415			19\$:	SCOP1			:CHECK IF LOOP ON ERROR
3977	021756	000241				CLC			:SHIFT IN ZERO
3978	021760	006137	002010			ROL	CONFIG		
3979	021764	005301				DEC	R1		:CHECK IF FINISHED
3980	021766	001402				BEQ	TST26		:YES, GO ON TO NEXT TEST
3981	021770	000137	021212			JMP	1\$		

\*\*\*\*\*  
 :\*TEST 26 REGISTER INTERACTION USING DISK ADDRESS (PART 2)  
 :\*

3982  
 3983  
 3984  
 3985



3986  
3987  
3988  
3989  
3990  
3991  
3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999 021774 000004  
4000 021776 012737 000764 001200  
4001 022004 012701 000021  
4002 022010 012737 177776 002010  
4003 022016 012737 064004 001320  
4004 022024 012762 100000 000000  
4005 022032 012737 022040 001110  
4006  
4007  
4008 022040  
4009 022040 005062 000002  
4010 022044 013762 002010 000006  
4011 022052 016237 000006 001126  
4012 022060 013737 002010 001124  
4013 022066 042737 174340 001124  
4014 022074 023737 001124 001126  
4015 022102 001404  
4016 022104 012737 066743 001322  
4017 022112 104003  
4018 022114  
4019 022114 016237 000000 001126  
4020 022122 022737 000200 001126  
4021 022130 001407  
4022 022132 012737 000100 001124  
4023 022140 012737 067376 001322  
4024 022146 104003  
4025 022150  
4026 022150 016237 000004 001126  
4027 022156 001406  
4028 022160 005037 001124  
4029 022164 012737 067447 001322  
4030 022172 104003  
4031 022174  
4032 022174 016237 000002 001126  
4033 022202 001406  
4034 022204 005037 001124  
4035 022210 012737 067417 001322  
4036 022216 104003  
4037 022220  
4038 022220 016237 000016 001126  
4039 022226 001406  
4040 022230 005037 001124  
4041 022234 012737 067640 001322

ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.  
WRITE THE WORD COUNT REGISTER WITH 0.  
WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH  
THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE  
SURE NO INTERACTION TAKES PLACE.  
177777 177767 177577 173777 077777  
177776 177757 177377 167777  
177775 177737 176777 157777  
177773 177677 175777 137777  
\*\*\*\*\*  
TST26: SCOPE  
MOV #500.,\$TIMES ;:DO 500. ITERATIONS  
MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS  
MOV #177776,CONFIG ;:LOAD INITIAL CONFIGURATION  
MOV #EM8,EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP  
1\$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG,RKDA(R2) ;:WRITE RKDA  
MOV RKDA(R2), \$BDDAT ;:STORE RKDA  
MOV CONFIG,\$GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #174340,\$GDDAT ;:INITIALIZE READ ONLY BITS  
CMP \$GDDAT,\$BDDAT ;:CHECK IF RKDA CORRECT  
BEQ 2\$ ;:YES,TEST IF ANY OTHER REG MODIFIED  
MOV #EM1003,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
2\$:  
MOV RKCS1(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY,\$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3\$ ;:YES, CONTINUE  
MOV #IR,\$GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
3\$:  
MOV RKBA(R2), \$BDDAT ;:STORE BUS AND REG  
BEQ 4\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
4\$:  
MOV RKWC(R2), \$BDDAT ;:STORE WORD COUNT REG  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
5\$:  
MOV RKASOF(R2), \$BDDAT ;:STORE ATTENTION SUMMARY/OFFSET REG.  
BEQ 7\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1024,EM3N+2 ;:LOAD ERROR MESSAGE







4098 022602 104415  
4099 022604 000261  
4100 022606 006137 002010  
4101 022612 005301  
4102 022614 001402  
4103 022616 000137 022040  
4104  
4105  
4106  
4107  
4108  
4109  
4110  
4111  
4112  
4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121 022622 000004  
4122 022624 012737 000764 001200  
4123 022632 012701 000021  
4124 022636 005037 002010  
4125 022642 012737 064004 001320  
4126 022650 012762 100000 000000  
4127 022656 012737 022664 001110  
4128  
4129  
4130 022664  
4131 022664 005062 000002  
4132 022670 013762 002010 000006  
4133 022676 016237 000006 001126  
4134 022704 013737 002010 001124  
4135 022712 042737 174340 001124  
4136 022720 023737 001124 001126  
4137 022726 001404  
4138 022730 012737 066743 001322  
4139 022736 104003  
4140 022740  
4141 022740 016237 000000 001126  
4142 022746 022737 000200 001126  
4143 022754 001407  
4144 022756 012737 000100 001124  
4145 022764 012737 067376 001322  
4146 022772 104003  
4147 022774  
4148 022774 016237 000004 001126  
4149 023002 001406  
4150 023004 005037 001124  
4151 023010 012737 067447 001322  
4152 023016 104003  
4153 023020

19\$: SCOPI ;CHECK IF LOOP ON ERROR  
SEC ;SHIFT IN ONE  
ROL CONFIG  
DEC R1 ;CHECK IF FINISHED  
BEQ TST27 ;:YES, GO ON TO NEXT TEST  
JMP 1\$

\*\*\*\*\*  
\*TEST 27 REGISTER INTERACTION USING DISK ADDRESS (PART 3)  
\*  
\* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.  
\* WRITE THE WORD COUNT REGISTER WITH 0.  
\*  
\* WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH  
\* THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE  
\* SURE NO INTERACTION TAKES PLACE.  
\*  
\* 000001 000037 000777 017777 000000  
\* 000003 000077 001777 037777  
\* 000007 000177 003777 077777  
\* 000017 000377 007777 177777  
\*\*\*\*\*

TST27: SCOPE  
MOV #500, \$TIMES ;:DO 500. ITERATIONS  
MOV #17, R1 ;LOAD NUMBER OF PATTERNS  
CLR CONFIG ;LOAD INITIAL CONFIGURATION  
MOV #EM8, EM3N ;LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MOV CONFIG, RKDA(R2) ;WRITE RKDA  
MOV RKDA(R2), \$BDDAT ;STORE RKDA  
MOV CONFIG, \$GDDAT ;PREPARE EXPECTED RESULTS  
BIC #174340, \$GDDAT ;INITIALIZE READ ONLY BITS  
CMP \$GDDAT, \$BDDAT ;CHECK IF RKDA CORRECT  
BEQ 2\$ ;YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1003, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

2\$: MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG. 1  
CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT  
BEQ 3\$ ;YES, CONTINUE  
MOV #IR, \$GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

3\$: MOV RKBA(R2), \$BDDAT ;STORE BUS AND REG  
BEQ 4\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

4\$:



4154	023020	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	;STORE WORD COUNT REG
4155	023026	001406				BEQ	5\$	;CHECK IF ZERO
4156	023030	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4157	023034	012737	067417	001322		MOV	#EM1018, EM3N+2	;LOAD ERROR MESSAGE
4158	023042	104003				ERROR	3	
4159	023044				5\$:			
4160	023044	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
4161	023052	001406				BEQ	7\$	;CHECK IF ZERO
4162	023054	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4163	023060	012737	067640	001322		MOV	#EM1024, EM3N+2	;LOAD ERROR MESSAGE
4164	023066	104003				ERROR	3	
4165	023070				7\$:			
4166	023070	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
4167	023076	022737	000100	001126		CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
4168	023104	001407				BEQ	8\$	;YES, CONTINUE
4169	023106	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
4170	023114	012737	067532	001322		MOV	#EM1021, EM3N+2	;LOAD ERROR MESSAGE
4171	023122	104003				ERROR	3	
4172	023124				8\$:			
4173	023124	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG
4174	023132	001406				BEQ	9\$	;CHECK IF ZERO
4175	023134	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4176	023140	012737	067553	001322		MOV	#EM1022, EM3N+2	;LOAD ERROR MESSAGE
4177	023146	104003				ERROR	3	
4178	023150	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
4179	023156	001406				BEQ	10\$	;CHECK IF ZERO
4180	023160	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4181	023164	012737	067611	001322		MOV	#EM1023, EM3N+2	;LOAD ERROR MESSAGE
4182	023172	104003				ERROR	3	
4183	023174				10\$:			
4184	023174	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
4185	023202	001406				BEQ	12\$	;CHECK IF ZERO
4186	023204	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4187	023210	012737	067706	001322		MOV	#EM1025, EM3N+2	;LOAD ERROR MESSAGE
4188	023216	104003				ERROR	3	
4189	023220				12\$:			
4190	023220	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG.1
4191	023226	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED MR1
4192	023234	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4193	023242	001403				BEQ	13\$	
4194	023244	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4195	023252	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
4196	023260	001404				BEQ	14\$	;YES, ISSUE CONTROLLER CLEAR
4197	023262	012737	067733	001322		MOV	#EM1026, EM3N+2	;LOAD ERROR MESSAGE
4198	023270	104003				ERROR	3	
4199	023272				14\$:			
4200	023272	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
4201	023300	001406				BEQ	15\$	;CHECK IF ZERO
4202	023302	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4203	023306	012737	070011	001322		MOV	#EM1030, EM3N+2	;LOAD ERROR MESSAGE
4204	023314	104003				ERROR	3	
4205	023316	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
4206	023324	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	;USE 4066
4207	023332	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
4208	023340	001404				BEQ	18\$	;YES, INITIALIZE RK611
4209	023342	012737	067764	001322		MOV	#EM1029, EM3N+2	;LOAD ERROR MESSAGE



```

4210 023350 104003          ERROR 3
4211 023352 016237 000006 002014 18$: MOV RKDA(R2),PREREG ;GET PREVIOUS CONTENTS
4212 023360 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
4213 023366 016237 000006 001126 MOV RKDA(R2),$BDDAT ;GET CURRENT VALUE
4214 023374 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4215 023400 023737 001124 001126 CMP $GDDAT,$BDDAT ;CHECK IF RKDA CORRECT
4216 023406 001407 BEQ 19$ ;YES, CHECK IF FINISHED
4217 023410 012737 063465 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
4218 023416 012737 066743 001332 MOV #EM1003,EM4N+2
4219 023424 104004          ERROR 4
4220 023426 104415          19$: SCOP1 ;CHECK IF LOOP ON ERROR
4221 023430 000261          SEC ;SHIFT IN ONE
4222 023432 006137 002010 ROL CONFIG
4223 023436 005301          DEC R1 ;CHECK IF FINISHED
4224 023440 001402          BEQ TST30 ;:YES, GO ON TO NEXT TEST
4225 023442 000137 022664          JMP 1$
  
```

```

*****
*TEST 30 REGISTER INTERACTION USING DISK ADDRESS (PART 4)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER WITH 0.
*
* WRITE THE DISK ADDRESS REGISTER (SECTOR AND TRACK) WITH
* THE FOLLOWING CONFIGURATIONS, CHECK REGISTER CONTENTS AND MAKE
* SURE NO INTERACTION TAKES PLACE.
*
* 100000 174000 177600 177770 000000
* 140000 176000 177700 177774
* 160000 177000 177740 177776
* 170000 177400 177760 177777
*
*****
  
```

```

4243 023446 000004          TST30: SCOPE
4244 023450 012737 000764 001200 MOV #500,$TIMES ;:DO 500. ITERATIONS
4245 023456 012701 000021 MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS
4246 023462 005037 002010 CLR CONFIG ;:LOAD INITIAL CONFIGURATION
4247 023466 012737 064004 001320 MOV #EM8,EM3N ;:LOAD ERROR MESSAGE
4248 023474 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR
4249 023502 012737 023510 001110 MOV #1$,$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR
4250 ; SUBTEST LOOP
4251
4252 023510          1$: CLR RKWC(R2) ;:CLEAR WORD COUNT REG.
4253 023510 005062 000002 MOV CONFIG,RKDA(R2) ;:WRITE RKDA
4254 023514 013762 002010 000006 MOV RKDA(R2),$BDDAT ;:STORE RKDA
4255 023522 016237 000006 001126 MOV CONFIG,$GDDAT ;:PREPARE EXPECTED RESULTS
4256 023530 013737 002010 001124 BIC #174340,$GDDAT ;:INITIALIZE READ ONLY BITS
4257 023536 042737 174340 001124 CMP $GDDAT,$BDDAT ;:CHECK IF RKDA CORRECT
4258 023544 023737 001124 001126 BEQ 2$ ;:YES,TEST IF ANY OTHER REG MODIFIED
4259 023552 001404 MOV #EM1003,EM3N+2 ;:LOAD ERROR MESSAGE
4260 023554 012737 066743 001322          ERROR 3
4261 023562 104003
4262 023564          2$: MOV RKCS1(R2),$BDDAT ;:STORE COMMAND AND STATUS REG. 1
4263 023564 016237 000000 001126 CMP #RDY,$BDDAT ;:CHECK IF CS1 CORRECT
4264 023572 022737 000200 001126 BEQ 3$ ;:YES, CONTINUE
4265 023600 001407
  
```



4266	023602	012737	000100	001124	MOV	#IR,\$GDDAT	;LOAD EXPECTED RESULTS
4267	023610	012737	067376	001322	MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
4268	023616	104003			ERROR	3	
4269	023620				3\$:		
4270	023620	016237	000004	001126	MOV	RKBA(R2),\$BDDAT	;STORE BUS AND REG
4271	023626	001406			BEQ	4\$	;CHECK IF ZERO
4272	023630	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4273	023634	012737	067447	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
4274	023642	104003			ERROR	3	
4275	023644				4\$:		
4276	023644	016237	000002	001126	MOV	RKWC(R2),\$BDDAT	;STORE WORD COUNT REG
4277	023652	001406			BEQ	5\$	;CHECK IF ZERO
4278	023654	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4279	023660	012737	067417	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
4280	023666	104003			ERROR	3	
4281	023670				5\$:		
4282	023670	016237	000016	001126	MOV	RKASOF(R2),\$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG.
4283	023676	001406			BEQ	7\$	;CHECK IF ZERO
4284	023700	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4285	023704	012737	067640	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
4286	023712	104003			ERROR	3	
4287	023714				7\$:		
4288	023714	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	;STORE COMMAND AND STATUS REG.2
4289	023722	022737	000100	001126	CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
4290	023730	001407			BEQ	8\$	;YES,CONTINUE
4291	023732	012737	000100	001124	MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
4292	023740	012737	067532	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
4293	023746	104003			ERROR	3	
4294	023750				8\$:		
4295	023750	016237	000012	001126	MOV	RKDS(R2),\$BDDAT	;STORE DRIVE STATUS REG
4296	023756	001406			BEQ	9\$	;CHECK IF ZERO
4297	023760	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4298	023764	012737	067553	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
4299	023772	104003			ERROR	3	
4300	023774	016237	000014	001126	MOV	RKER(R2),\$BDDAT	;STORE ERROR REG
4301	024002	001406			BEQ	10\$	;CHECK IF ZERO
4302	024004	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4303	024010	012737	067611	001322	MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
4304	024016	104003			ERROR	3	
4305	024020				10\$:		
4306	024020	016237	000020	001126	MOV	RKDCYL(R2),\$BDDAT	;STORE CYLINDER ADD REG
4307	024026	001406			BEQ	12\$	;CHECK IF ZERO
4308	024030	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
4309	024034	012737	067706	001322	MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
4310	024042	104003			ERROR	3	
4311	024044				12\$:		
4312	024044	016237	000026	001126	MOV	RKMR1(R2),\$BDDAT	;STORE MAINTENANCE REG.1
4313	024052	012737	002000	001124	MOV	#MEWD,\$GDDAT	;LOAD EXPECTED MR1
4314	024060	032737	02C000	001126	BIT	#ECCW,\$BDDAT	
4315	024066	001403			BEQ	13\$	
4316	024070	052737	020000	001124	BIS	#ECCW,\$GDDAT	
4317	024076	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	;CHECK IF MR1 CORRECT
4318	024104	001404			BEQ	14\$	;YES,ISSUE CONTROLLER CLEAR
4319	024106	012737	067733	001322	MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
4320	024114	104003			ERROR	3	
4321	024116				14\$:		



```

4322 024116 016237 000032 001126      MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
4323 024124 001406                BEQ      15$                ;CHECK IF ZERO
4324 024126 005037 001124          CLR      $GDDAT            ;LOAD EXPECTED CONTENTS
4325 024132 012737 070011 001322      MOV      #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
4326 024140 104003                ERROR    3
4327 024142 016237 000030 001126 15$:    MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
4328 024150 012737 004066 001124 16$:    MOV      #4066, $GDDAT     ;USE 4066
4329 024156 023737 001124 001126 17$:    CMP      $GDDAT, $BDDAT   ;CHECK IF ECC POSITION CORRECT
4330 024164 001404                BEQ      18$                ;YES, INITIALIZE RK611
4331 024166 012737 067764 001322      MOV      #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
4332 024174 104003                ERROR    3
4333 024176 016237 000006 002014 18$:    MOV      RKDA(R2), PREREG ;GET PREVIOUS CONTENTS
4334 024204 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
4335 024212 016237 000006 001126      MOV      RKDA(R2), $BDDAT ;GET CURRENT VALUE
4336 024220 005037 001124          CLR      $GDDAT            ;LOAD EXPECTED CONTENTS
4337 024224 023737 001124 001126      CMP      $GDDAT, $BDDAT   ;CHECK IF RKDA CORRECT
4338 024232 001407                BEQ      19$                ;YES, CHECK IF FINISHED
4339 024234 012737 063465 001330      MOV      #EM3, EM4N       ;LOAD ERROR MESSAGE
4340 024242 012737 066743 001332      MOV      #EM1003, EM4N+2
4341 024250 104004                ERROR    4
4342 024252 104415                19$:    SCOP1                    ;CHECK IF LOOP ON ERROR
4343 024254 000261                SEC                        ;SHIFT IN ONE
4344 024256 006137 002010          ROL      CONFIG
4345 024262 005301                DEC      R1                ;CHECK IF FINISHED
4346 024264 001402                BEQ      TST31             ;:YES, GO ON TO NEXT TEST
4347 024266 000137 023510          JMP      1$

```

```

*****
:TEST 31 REGISTER INTERACTION USING ATTN/OFFSET (PART 1)
:
:ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
:WRITE THE WORD COUNT REGISTER WITH 0.
:
:WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
:WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
:MAKE SURE NO INTERACTION TAKES PLACE.
:
:000000 000010 000200 004000 100000
:000001 000020 000400 010000
:000002 000040 001000 020000
:000004 000100 002000 040000

```

```

4365 024272 000004                TST31: SCOPE
4366 024274 012737 000764 001200      MOV      #500., $TIMES    ;;DO 500. ITERATIONS
4367 024302 012701 000021          MOV      #17., R1        ;LOAD NUMBER OF PATTERNS
4368 024306 012737 000001 002010      MOV      #000001, CONFIG ;LOAD INITIAL CONFIGURATION
4369 024314 012737 064433 001320      MOV      #EM17, EM3N     ;LOAD ERROR MESSAGE
4370 024322 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4371 024330 012737 024336 001110      MOV      #1$, $LPERR     ;LOAD LOOP ON ERROR LOCATION FOR
4372                                ; SUBTEST LOOP
4373
4374 024336                1$:
4375 024336 005062 000002          CLR      RKWC(R2)        ;CLEAR WORD COUNT REG.
4376 024342 013762 002010 000016      MOV      CONFIG, RKASOF(R2) ;WRITE RKASOF
4377 024350 016237 000016 001126      MOV      RKASOF(R2), $BDDAT ;STORE RKASOF

```



4378	024356	013737	002010	001124	MOV	CONFIG,\$GDDAT	:PREPARE EXPECTED RESULTS
4379	024364	042737	177400	001124	BIC	#177400,\$GDDAT	:INITIALIZE READ ONLY BITS
4380	024372	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKASOF CORRECT
4381	024400	001404			BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
4382	024402	012737	067063	001322	MOV	#EM1007,EM3N+2	:LOAD ERROR MESSAGE
4383	024410	104003			ERROR	3	
4384	024412				2\$:		
4385	024412	016237	000000	001126	MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
4386	024420	022737	000200	001126	CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
4387	024426	001407			BEQ	3\$	:YES, CONTINUE
4388	024430	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
4389	024436	012737	067376	001322	MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
4390	024444	104003			ERROR	3	
4391	024446				3\$:		
4392	024446	016237	000004	001126	MOV	RKBA(R2),\$BDDAT	:STORE BUS AND REG
4393	024454	001406			BEQ	4\$	:CHECK IF ZERO
4394	024456	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4395	024462	012737	067447	001322	MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
4396	024470	104003			ERROR	3	
4397	024472				4\$:		
4398	024472	016237	000002	001126	MOV	RKWC(R2),\$BDDAT	:STORE WORD COUNT REG
4399	024500	001406			BEQ	5\$	:CHECK IF ZERO
4400	024502	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4401	024506	012737	067417	001322	MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
4402	024514	104003			ERROR	3	
4403	024516				5\$:		
4404	024516	016237	000006	001126	MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
4405	024524	001406			BEQ	6\$	:CHECK IF ZERO
4406	024526	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4407	024532	012737	067474	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
4408	024540	104003			ERROR	3	
4409	024542				6\$:		
4410	024542	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
4411	024550	022737	000100	001126	CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
4412	024556	001407			BEQ	8\$	:YES,CONTINUE
4413	024560	012737	000100	001124	MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
4414	024566	012737	067532	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
4415	024574	104003			ERROR	3	
4416	024576				8\$:		
4417	024576	016237	000012	001126	MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
4418	024604	001406			BEQ	9\$	:CHECK IF ZERO
4419	024606	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4420	024612	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
4421	024620	104003			ERROR	3	
4422	024622	016237	000014	001126	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
4423	024630	001406			BEQ	10\$	:CHECK IF ZERO
4424	024632	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4425	024636	012737	067611	001322	MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
4426	024644	104003			ERROR	3	
4427	024646				10\$:		
4428	024646	016237	000020	001126	MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
4429	024654	001406			BEQ	12\$	:CHECK IF ZERO
4430	024656	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4431	024662	012737	067706	001322	MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
4432	024670	104003			ERROR	3	
4433	024672				12\$:		



```

4434 024672 016237 000026 001126 MOV RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG.1
4435 024700 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MR1
4436 024706 032737 020000 001126 BIT #ECCW, $BDDAT
4437 024714 001403 BEQ 13$
4438 024716 052737 020000 001124 BIS #ECCW, $GDDAT
4439 024724 023737 001124 001126 13$: CMP $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
4440 024732 001404 BEQ 14$ ;YES, ISSUE CONTROLLER CLEAR
4441 024734 012737 067733 001322 MOV #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
4442 024742 104003 ERROR 3
4443 024744 14$:
4444 024744 016237 000032 001126 MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
4445 024752 001406 BEQ 15$ ;CHECK IF ZERO
4446 024754 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4447 024760 012737 070011 001322 MOV #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
4448 024766 104003 ERROR 3
4449 024770 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
4450 024776 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
4451 025004 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF FCC POSITION CORRECT
4452 025012 001404 BEQ 18$ ;YES, INITIALIZE RK611
4453 025014 012737 067764 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
4454 025022 104003 ERROR 3
4455 025024 016237 000016 002014 18$: MOV RKASOF(R2), PREREG ;GET PREVIOUS CONTENTS
4456 025032 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
4457 025040 016237 000016 001126 MOV RKASOF(R2), $BDDAT ;GET CURRENT VALUE
4458 025046 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
4459 025052 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKASOF CORRECT
4460 025060 001407 BEQ 19$ ;YES, CHECK IF FINISHED
4461 025062 012737 063465 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
4462 025070 012737 067063 001332 MOV #EM1007, EM4N+2
4463 025076 104004 ERROR 4
4464 025100 104415 19$: SCOPI ;CHECK IF LOOP ON ERROR
4465 025102 000241 CLC ;SHIFT IN ZERO
4466 025104 006137 002010 ROL CONFIG
4467 025110 005301 DEC R1 ;CHECK IF FINISHED
4468 025112 001402 BEQ TST32 ;:YES, GO ON TO NEXT TEST
4469 025114 000137 024336 JMP 1$
    
```

```

4470
4471
4472 :*****
4473 :*TEST 32 REGISTER INTERACTION USING ATTN/OFFSET (PART 2)
4474 :*
4475 :* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4476 :* WRITE THE WORD COUNT REGISTER WITH 0.
4477 :*
4478 :* WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
4479 :* WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
4480 :* MAKE SURE NO INTERACTION TAKES PLACE.
4481 :*
4482 :* 177777 177767 177577 173777 077777
4483 :* 177776 177757 177377 167777
4484 :* 177775 177737 176777 157777
4485 :* 177773 177677 175777 137777
4486 :*
4487 :*****
    
```

```

4487 025120 000004 TST32: SCOPE
4488 025122 012737 000764 001200 MOV #500, $TIMES ;:DO 500. ITERATIONS
4489 025130 012701 000021 MOV #17, R1 ;LOAD NUMBER OF PATTERNS
    
```



4490	025134	012737	177776	002010		MOV	#177776,CONFIG	:LOAD INITIAL CONFIGURATION
4491	025142	012737	064433	001320		MOV	#EM17,EM3N	:LOAD ERROR MESSAGE
4492	025150	012762	100000	000000		MOV	#CCLR,RKCS1(R2)	:CLEAR RK611 WITH CONTROLLER CLEAR
4493	025156	012737	025164	001110		MOV	#1\$,SLFERR	:LOAD LOOP ON ERROR LOCATION FOR
4494								: SUBTEST LOOP
4495								
4496	025164				1\$:			
4497	025164	005062	000002			CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
4498	025170	013762	002010	000016		MOV	CONFIG,RKASOF(R2)	:WRITE RKASOF
4499	025176	016237	000016	001126		MOV	RKASOF(R2),\$BDDAT	:STORE RKASOF
4500	025204	013737	002010	001124		MOV	CONFIG,\$GDDAT	:PREPARE EXPECTED RESULTS
4501	025212	042737	177400	001124		BIC	#177400,\$GDDAT	:INITIALIZE READ ONLY BITS
4502	025220	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	:CHECK IF RKASOF CORRECT
4503	025226	001404				BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
4504	025230	012737	067063	001322		MOV	#EM1007,EM3N+2	:LOAD ERROR MESSAGE
4505	025236	104003				ERROR	3	
4506	025240				2\$:			
4507	025240	016237	000000	001126		MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG. 1
4508	025246	022737	000200	001126		CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
4509	025254	001407				BEQ	3\$	:YES, CONTINUE
4510	025256	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED RESULTS
4511	025264	012737	067376	001322		MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
4512	025272	104003				ERROR	3	
4513	025274				3\$:			
4514	025274	016237	000004	001126		MOV	RKBA(R2),\$BDDAT	:STORE BUS AND REG
4515	025302	001406				BEQ	4\$	:CHECK IF ZERO
4516	025304	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4517	025310	012737	067447	001322		MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
4518	025316	104003				ERROR	3	
4519	025320				4\$:			
4520	025320	016237	000002	001126		MOV	RKWC(R2),\$BDDAT	:STORE WORD COUNT REG
4521	025326	001406				BEQ	5\$	:CHECK IF ZERO
4522	025330	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4523	025334	012737	067417	001322		MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
4524	025342	104003				ERROR	3	
4525	025344				5\$:			
4526	025344	016237	000006	001126		MOV	RKDA(R2),\$BDDAT	:STORE DISK AVERAGE REG
4527	025352	001406				BEQ	6\$	:CHECK IF ZERO
4528	025354	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4529	025360	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
4530	025366	104003				ERROR	3	
4531	025370				6\$:			
4532	025370	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.2
4533	025376	022737	000100	001126		CMP	#IR,\$BDDAT	:CHECK IF CS2 CORRECT
4534	025404	001407				BEQ	8\$	:YES,CONTINUE
4535	025406	012737	000100	001124		MOV	#IR,\$GDDAT	:LOAD EXPECTED CONTENTS
4536	025414	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
4537	025422	104003				ERROR	3	
4538	025424				8\$:			
4539	025424	016237	000012	001126		MOV	RKDS(R2),\$BDDAT	:STORE DRIVE STATUS REG
4540	025432	001406				BEQ	9\$	:CHECK IF ZERO
4541	025434	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4542	025440	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
4543	025446	104003				ERROR	3	
4544	025450	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REG
4545	025456	001406				BEQ	10\$	:CHECK IF ZERO



```

4546 025460 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
4547 025464 012737 067611 001322  MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
4548 025472 104003                ERROR    3
4549 025474                10$:
4550 025474 016237 000020 001126  MOV      RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
4551 025502 001406                BEQ      12$              ;CHECK IF ZERO
4552 025504 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
4553 025510 012737 067706 001322  MOV      #EM1025,EM3N+2 ;LOAD ERROR MESSAGE
4554 025516 104003                ERROR    3
4555 025520                12$:
4556 025520 016237 000026 001126  MOV      RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
4557 025526 012737 002000 001124  MOV      #MEWD, $GDDAT    ;LOAD EXPECTED MR1
4558 025534 032737 020000 001126  BIT      #ECCW, $BDDAT
4559 025542 001403                BEQ      13$              ;CHECK IF MR1 CORRECT
4560 025544 052737 020000 001124  BIS      #ECCW, $GDDAT    ;YES, ISSUE CONTROLLER CLEAR
4561 025552 023737 001124 001126  CMP      $GDDAT, $BDDAT
4562 025560 001404                BEQ      14$              ;LOAD ERROR MESSAGE
4563 025562 012737 067733 001322  MOV      #EM1026,EM3N+2
4564 025570 104003                ERROR    3
4565 025572                14$:
4566 025572 016237 000032 001126  MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
4567 025600 001406                BEQ      15$              ;CHECK IF ZERO
4568 025602 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
4569 025606 012737 070011 001322  MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
4570 025614 104003                ERROR    3
4571 025616 016237 000030 001126 15$: MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
4572 025624 012737 004066 001124 16$: MOV      #4066, $GDDAT    ;USE 4066
4573 025632 023737 001124 001126 17$: CMP      $GDDAT, $BDDAT  ;CHECK IF ECC POSITION CORRECT
4574 025640 001404                BEQ      18$              ;YES, INITIALIZE RK611
4575 025642 012737 067764 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
4576 025650 104003                ERROR    3
4577 025652 016237 000016 002014 18$: MOV      RKASOF(R2), PREREG ;GET PREVIOUS CONTENTS
4578 025660 012762 100000 000000  MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
4579 025666 016237 000016 001126  MOV      RKASOF(R2), $BDDAT ;GET CURRENT VALUE
4580 025674 005037 001124          CLR      $GDDAT          ;LOAD EXPECTED CONTENTS
4581 025700 023737 001124 001126  CMP      $GDDAT, $BDDAT  ;CHECK IF RKASOF CORRECT
4582 025706 001407                BEQ      19$              ;YES, CHECK IF FINISHED
4583 025710 012737 063465 001330  MOV      #EM3, EM4N      ;LOAD ERROR MESSAGE
4584 025716 012737 067063 001332  MOV      #EM1007,EM4N+2
4585 025724 104004                ERROR    4
4586 025726 104415                19$: SCOP1                ;CHECK IF LOOP ON ERROR
4587 025730 000261                SEC                  ;SHIFT IN ONE
4588 025732 006137 002010  ROL      CONFIG
4589 025736 005301                DEC      R1              ;CHECK IF FINISHED
4590 025740 001402                BEQ      TST33           ;:YES, GO ON TO NEXT TEST
4591 025742 000137 025164                JMP      1$

```

```

4592
4593 *****
4594 *TEST 33 REGISTER INTERACTION USING ATTN/OFFSET (PART 3)
4595 *
4596 * ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.
4597 * WRITE THE WORD COUNT REGISTER WITH 0.
4598 *
4599 * WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER
4600 * WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND
4601 * MAKE SURE NO INTERACTION TAKES PLACE.

```



```
4602
4603
4604
4605
4606
4607
4608
4609 025746 000004
4610 025750 012737 000764 001200
4611 025756 012701 000021
4612 025762 005037 002010
4613 025766 012737 064433 001320
4614 025774 012762 100000 000000
4615 026002 012737 026010 001110
4616
4617
4618 026010
4619 026010 005062 000002
4620 026014 013762 002010 000016
4621 026022 016237 000016 001126
4622 026030 013737 002010 001124
4623 026036 042737 177400 001124
4624 026044 023737 001124 001126
4625 026052 001404
4626 026054 012737 067063 001322
4627 026062 104003
4628 026064
4629 026064 016237 000000 001126
4630 026072 022737 000200 001126
4631 026100 001407
4632 026102 012737 000100 001124
4633 026110 012737 067376 001322
4634 026116 104003
4635 026120
4636 026120 016237 000004 001126
4637 026126 001406
4638 026130 005037 001124
4639 026134 012737 067447 001322
4640 026142 104003
4641 026144
4642 026144 016237 000002 001126
4643 026152 001406
4644 026154 005037 001124
4645 026160 012737 067417 001322
4646 026166 104003
4647 026170
4648 026170 016237 000006 001126
4649 026176 001406
4650 026200 005037 001124
4651 026204 012737 067474 001322
4652 026212 104003
4653 026214
4654 026214 016237 000010 001126
4655 026222 022737 000100 001126
4656 026230 001407
4657 026232 012737 000100 001124
```

\*\*\*\*\*  
TST33: SCOPE  
MOV #500.,\$TIMES ;:DO 500. ITERATIONS  
MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS  
CLR CONFIG ;:LOAD INITIAL CONFIGURATION  
MOV #EM17,EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG,RKASOF(R2) ;:WRITE RKASOF  
MOV RKASOF(R2), \$BDDAT ;:STORE RKASOF  
MOV CONFIG,\$GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #177400,\$GDDAT ;:INITIALIZE READ ONLY BITS  
CMP \$GDDAT,\$BDDAT ;:CHECK IF RKASOF CORRECT  
BEQ 2\$ ;:YES,TEST IF ANY OTHER REG MODIFIED  
MOV #EM1007,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

2\$:  
MOV RKCS1(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY,\$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3\$ ;:YES, CONTINUE  
MOV #IR,\$GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

3\$:  
MOV RKBA(R2), \$BDDAT ;:STORE BUS AND REG  
BEQ 4\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

4\$:  
MOV RKWC(R2), \$BDDAT ;:STORE WORD COUNT REG  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

5\$:  
MOV RKDA(R2), \$BDDAT ;:STORE DISK AVERAGE REG  
BEQ 6\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1020,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

6\$:  
MOV RKCS2(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG.2  
CMP #IR,\$BDDAT ;:CHECK IF CS2 CORRECT  
BEQ 8\$ ;:YES, CONTINUE  
MOV #IR,\$GDDAT ;:LOAD EXPECTED CONTENTS



4658	026240	012737	067532	001322		MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
4659	026246	104003				ERROR	3	
4660	026250				8\$:			
4661	026250	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
4662	026256	001406				BEQ	9\$	:CHECK IF ZERO
4663	026260	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4664	026264	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
4665	026272	104003				ERROR	3	
4666	026274	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
4667	026302	001406				BEQ	10\$	:CHECK IF ZERO
4668	026304	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4669	026310	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
4670	026316	104003				ERROR	3	
4671	026320				10\$:			
4672	026320	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
4673	026326	001406				BEQ	12\$	:CHECK IF ZERO
4674	026330	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4675	026334	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
4676	026342	104003				ERROR	3	
4677	026344				12\$:			
4678	026344	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
4679	026352	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
4680	026360	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4681	026366	001403				BEQ	13\$	
4682	026370	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4683	026376	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
4684	026404	001404				BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
4685	026406	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
4686	026414	104003				ERROR	3	
4687	026416				14\$:			
4688	026416	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
4689	026424	001406				BEQ	15\$	:CHECK IF ZERO
4690	026426	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4691	026432	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
4692	026440	104003				ERROR	3	
4693	026442	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
4694	026450	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
4695	026456	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
4696	026464	001404				BEQ	18\$	:YES, INITIALIZE RK611
4697	026466	012737	067764	001322		MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
4698	026474	104003				ERROR	3	
4699	026476	016237	000016	002014	18\$:	MOV	RKASOF(R2), PREREG	:GET PREVIOUS CONTENTS
4700	026504	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
4701	026512	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	:GET CURRENT VALUE
4702	026520	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4703	026524	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	:CHECK IF RKASOF CORRECT
4704	026532	001407				BEQ	19\$	:YES, CHECK IF FINISHED
4705	026534	012737	063465	001330		MOV	#EM3, EM4N	:LOAD ERROR MESSAGE
4706	026542	012737	067063	001332		MOV	#EM1007,EM4N+2	
4707	026550	104004				ERROR	4	
4708	026552	104415			19\$:	SCOPI		:CHECK IF LOOP ON ERROR
4709	026554	000261				SEC		:SHIFT IN ONE
4710	026556	006137	002010			ROL	CONFIG	
4711	026562	005301				DEC	R1	:CHECK IF FINISHED
4712	026564	001402				BEQ	TST34	:YES, GO ON TO NEXT TEST
4713	026566	000137	026010			JMP	1\$	



4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757  
4758  
4759  
4760  
4761  
4762  
4763  
4764  
4765  
4766  
4767  
4768  
4769

026572	000004			
026574	012737	000764	001200	
026602	012701	000021		
026606	005037	002010		
026612	012737	064433	001320	
026620	012762	100000	000000	
026626	012737	026634	001110	
026634				
026634	005062	000002		
026640	013762	002010	000016	
026646	016237	000016	001126	
026654	013737	002010	001124	
026662	042737	177400	001124	
026670	023737	001124	001126	
026676	001404			
026700	012737	067063	001322	
026706	104003			
026710				
026710	016237	000000	001126	
026716	022737	000200	001126	
026724	001407			
026726	012737	000100	001124	
026734	012737	067376	001322	
026742	104003			
026744				
026744	016237	000004	001126	
026752	001406			
026754	005037	001124		
026760	012737	067447	001322	
026766	104003			
026770				
026770	016237	000002	001126	
026776	001406			
027000	005037	001124		
027004	012737	067417	001322	
027012	104003			
027014				

```
*****  
*TEST 34 REGISTER INTERACTION USING ATTN/OFFSET (PART 4)  
*  
* ISSUE A CONTROLLER CLEAR TO INITIALIZE THE RK611 REGISTERS.  
* WRITE THE WORD COUNT REGISTER WITH 0.  
*  
* WRITE THE ATTENTION SUMMARY AND OFFSET REGISTER  
* WITH THE FOLLOWING CONFIGURATIONS. CHECK REGISTER CONTENTS AND  
* MAKE SURE NO INTERACTION TAKES PLACE.  
*  
* 100000 174000 177600 177770 000000  
* 140000 176000 177700 177774  
* 160000 177000 177740 177776  
* 170000 177400 177760 177777  
*****
```

```
TST34: SCOPE  
MOV #500,$TIMES ;;DO 500. ITERATIONS  
MOV #17,R1 ;LOAD NUMBER OF PATTERNS  
CLR CONFIG ;LOAD INITIAL CONFIGURATION  
MOV #EM17,EM3N ;LOAD ERROR MESSAGE  
MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP  
  
1$:  
CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MOV CONFIG,RKASOF(R2) ;WRITE RKASOF  
MOV RKASOF(R2),$BDDAT ;STORE RKASOF  
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS  
BIC #177400,$GDDAT ;INITIALIZE READ ONLY BITS  
CMP $GDDAT,$BDDAT ;CHECK IF RKASOF CORRECT  
BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED  
MOV #EM1007,EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
  
2$:  
MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1  
CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT  
BEQ 3$ ;YES, CONTINUE  
MOV #IR,$GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
  
3$:  
MOV RKBA(R2),$BDDAT ;STORE BUS AND REG  
BEQ 4$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
  
4$:  
MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG  
BEQ 5$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
  
5$:
```



4770	027014	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
4771	027022	001406				BEQ	6\$	:CHECK IF ZERO
4772	027024	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4773	027030	012737	067474	001322		MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
4774	027036	104003				ERROR	3	
4775	027040				6\$:			
4776	027040	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
4777	027046	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
4778	027054	001407				BEQ	8\$	:YES, CONTINUE
4779	027056	012737	000100	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
4780	027064	012737	067532	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
4781	027072	104003				ERROR	3	
4782	027074				8\$:			
4783	027074	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
4784	027102	001406				BEQ	9\$	:CHECK IF ZERO
4785	027104	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4786	027110	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
4787	027116	104003				ERROR	3	
4788	027120	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
4789	027126	001406				BEQ	10\$	:CHECK IF ZERO
4790	027130	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4791	027134	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
4792	027142	104003				ERROR	3	
4793	027144				10\$:			
4794	027144	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
4795	027152	001406				BEQ	12\$	:CHECK IF ZERO
4796	027154	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4797	027160	012737	067706	001322		MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
4798	027166	104003				ERROR	3	
4799	027170				12\$:			
4800	027170	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
4801	027176	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
4802	027204	032737	020000	001126		BIT	#ECCW, \$BDDAT	
4803	027212	001403				BEQ	13\$	
4804	027214	052737	020000	001124		BIS	#ECCW, \$GDDAT	
4805	027222	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
4806	027230	001404				BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
4807	027232	012737	067733	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
4808	027240	104003				ERROR	3	
4809	027242				14\$:			
4810	027242	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
4811	027250	001406				BEQ	15\$	:CHECK IF ZERO
4812	027252	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4813	027256	012737	070011	001322		MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
4814	027264	104003				ERROR	3	
4815	027266	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
4816	027274	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
4817	027302	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
4818	027310	001404				BEQ	18\$	:YES, INITIALIZE RK611
4819	027312	012737	067764	001322		MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
4820	027320	104003				ERROR	3	
4821	027322	016237	000016	002014	18\$:	MOV	RKASOF(R2), PREREG	:GET PREVIOUS CONTENTS
4822	027330	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
4823	027336	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	:GET CURRENT VALUE
4824	027344	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4825	027350	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	:CHECK IF RKASOF CORRECT



```

4826 027356 001407 BEQ 19$ ;YES, CHECK IF FINISHED
4827 027360 012737 063465 001330 MOV #EM3,EM4N ;LOAD ERROR MESSAGE
4828 027366 012737 067063 001332 MOV #EM1007,EM4N+2
4829 027374 104004 ERROR 4
4830 027376 104415 19$: SCOP1 ;CHECK IF LOOP ON ERROR
4831 027400 000261 SEC ;SHIFT IN ONE
4832 027402 006037 002010 ROR CONFIG
4833 027406 005301 DEC R1 ;CHECK IF FINISHED
4834 027410 001402 BEQ TST35 ;:YES, GO ON TO NEXT TEST
4835 027412 000137 026634 JMP 1$
  
```

\*\*\*\*\*  
 \*TEST 35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1)  
 \*\*\*\*\*

```

*
* RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
* WRITE THE WORD COUNT REGISTER WITH ZERO.
*
* WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
* SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
* REG. 2 AND CHECK FOR REGISTER INTERACTION.
*
* 000000 000010 000400 010000
* 000001 000020 001000 020000
* 000002 000100 002000 040000
* 000004 000200 004000 100000
  
```

```

4853 027416 000004 TST35: SCOPE
4854 027420 012737 000764 001200 MOV #500, $TIMES ;;DO 500. ITERATIONS
4855 027426 012701 000021 MOV #17, R1 ;LOAD NUMBER OF PATTERNS
4856 027432 012737 000001 002010 MOV #000001, CONFIG ;LOAD INITIAL CONFIGURATION
4857 027440 012737 000001 002012 MOV #000001, CONFIG1
4858 027446 012737 064510 001320 MOV #EM18, EM3N ;LOAD ERROR MESSAGE
4859 027454 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
4860 027462 012737 027470 001110 MOV #1$, $LPERR ;LOAD LOOP ON ERROR LOCATION FOR
4861 ; SUBTEST LOOP
4862
4863 027470 19$:
4864 027470 005062 000002 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
4865 027474 013762 002010 000010 MOV CONFIG, RKCS2(R2) ;WRITE RKCS2
4866 027502 016237 000010 001126 MOV RKCS2(R2), $BDDAT ;STORE RKCS2
4867 027510 013737 002010 001124 MOV CONFIG, $GDDAT ;PREPARE EXPECTED RESULTS
4868 027516 042737 177600 001124 BIC #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR, $GDDAT
4869 027524 052737 000100 001124 BIS #IR, $GDDAT ;INITIALIZE READ ONLY BITS
4870 027532 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKCS2 CORRECT
4871 027540 001404 BEQ 2$ ;YES, TEST IF ANY OTHER REG MODIFIED
4872 027542 012737 066766 001322 MOV #EM1004, EM3N+2 ;LOAD ERROR MESSAGE
4873 027550 104003 ERROR 3
4874 027552
4875 027552 016237 000000 001126 2$: MOV RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 1
4876 027560 022737 000200 001126 CMP #RDY, $BDDAT ;CHECK IF CS1 CORRECT
4877 027566 001407 BEQ 3$ ;YES, CONTINUE
4878 027570 012737 000100 001124 MOV #IR, $GDDAT ;LOAD EXPECTED RESULTS
4879 027576 012737 067376 001322 MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE
4880 027604 104003 ERROR 3
4881 027606 3$:
  
```



B 8

CZR6ADO RK611 DSKLS CTRL PRT1 MACY11 30(1046) 14-SEP-81 15:04 PAGE 93  
 CZR6AD.P11 14-SEP-81 13:43 T35 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 1) SEQ 0092

4882	027606	016237	000004	001126	MOV	RKBA(R2), \$BDDAT	:STORE BUS AND REG
4883	027614	001406			BEQ	4\$	:CHECK IF ZERO
4884	027616	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4885	027622	012737	067447	001322	MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
4886	027630	104003			ERROR	3	
4887	027632				4\$:		
4888	027632	016237	000002	001126	MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG
4889	027640	001406			BEQ	5\$	:CHECK IF ZERO
4890	027642	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4891	027646	012737	067417	001322	MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
4892	027654	104003			ERROR	3	
4893	027656				5\$:		
4894	027656	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
4895	027664	001406			BEQ	6\$	:CHECK IF ZERO
4896	027666	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4897	027672	012737	067474	001322	MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
4898	027700	104003			ERROR	3	
4899	027702				6\$:		
4900	027702	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
4901	027710	001406			BEQ	7\$	:CHECK IF ZERO
4902	027712	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4903	027716	012737	067640	001322	MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
4904	027724	104003			ERROR	3	
4905	027726				7\$:		
4906	027726	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
4907	027734	001406			BEQ	9\$	:CHECK IF ZERO
4908	027736	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4909	027742	012737	067553	001322	MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
4910	027750	104003			ERROR	3	
4911	027752	016237	000014	001126	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
4912	027760	001406			BEQ	10\$	:CHECK IF ZERO
4913	027762	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4914	027766	012737	067611	001322	MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
4915	027774	104003			ERROR	3	
4916	027776				10\$:		
4917	027776	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
4918	030004	001406			BEQ	12\$	:CHECK IF ZERO
4919	030006	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4920	030012	012737	067706	001322	MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
4921	030020	104003			ERROR	3	
4922	030022				12\$:		
4923	030022	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
4924	030030	012737	002000	001124	MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
4925	030036	032737	020000	001126	BIT	#ECCW, \$BDDAT	
4926	030044	001403			BEQ	13\$	
4927	030046	052737	020000	001124	BIS	#ECCW, \$GDDAT	
4928	030054	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT :CHECK IF MR1 CORRECT
4929	030062	001404			BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
4930	030064	012737	067733	001322	MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
4931	030072	104003			ERROR	3	
4932	030074				14\$:		
4933	030074	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
4934	030102	001406			BEQ	15\$	:CHECK IF ZERO
4935	030104	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
4936	030110	012737	070011	001322	MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
4937	030116	104003			ERROR	3	



```

4938 030120 016237 000030 001126 15$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
4939 030126 012737 004066 001124 16$: MOV #4066, $GDDAT ;USE 4066
4940 030134 023737 001124 001126 17$: CMP $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
4941 030142 001404 BEQ 18$ ;YES, INITIALIZE RK611
4942 030144 012737 067764 001322 MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
4943 030152 104003 ERROR 3
4944 030154 016237 000010 002014 18$: MOV RKCS2(R2), PREREG ;GET PREVIOUS CONTENTS
4945 030162 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
4946 030170 016237 000010 001126 MOV RKCS2(R2), $BDDAT ;GET CURRENT VALUE
4947 030176 012737 000100 001124 MOV #IR, $GDDAT ;LOAD EXPECTED CONTENTS
4948 030204 023737 001124 001126 CMP $GDDAT, $BDDAT ;CHECK IF RKCS2 CORRECT
4949 030212 001407 BEQ 19$ ;YES, CHECK IF FINISHED
4950 030214 012737 063465 001330 MOV #EM3, EM4N ;LOAD ERROR MESSAGE
4951 030222 012737 066766 001332 MOV #EM1004, EM4N+2
4952 030230 104004 ERROR 4
4953 030232 104415 19$: SCOP1 ;CHECK IF LOOP ON ERROR
4954 030234 000241 CLC ;SHIFT IN ZERO
4955 030236 006137 002012 ROL CONFIG1
4956 030242 013737 002012 002010 MOV CONFIG1, CONFIG ;MAKE SURE SUBSYSTEM CLEAR
4957 030250 042737 000040 002010 BIC #SCLR, CONFIG ; DOES NOT SET
4958 030256 005301 DEC R1 ;CHECK IF FINISHED
4959 030260 001402 BEQ TST36 ;:YES, GO ON TO NEXT TEST
4960 030262 000137 027470 JMP 1$
    
```

```

4961
4962
4963 *****
4964 *TEST 36 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2)
4965 *
4966 * RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
4967 * WRITE THE WORD COUNT REGISTER WITH ZERO.
4968 *
4969 * WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
4970 * SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
4971 * REG. 2 AND CHECK FOR REGISTER INTERACTION.
4972 *
4973 * 177737 177727 177337 167737
4974 * 177736 177717 176737 157737
4975 * 177735 177637 175737 137737
4976 * 177733 177537 173737 077737
4977 *
    
```

```

4978 030266 000004 TST36: SCOPE
4979 030270 012737 000764 001200 MOV #500, $TIMES ;:DO 500. ITERATIONS
4980 030276 012701 000021 MOV #17, R1 ;:LOAD NUMBER OF PATTERNS
4981 030302 012737 177736 002010 MOV #177736, CONFIG ;:LOAD INITIAL CONFIGURATION
4982 030310 012737 177776 002012 MOV #177776, CONFIG1
4983 030316 012737 064510 001320 MOV #EM18, EM3N ;:LOAD ERROR MESSAGE
4984 030324 012762 100000 000000 MOV #CCLR, RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR
4985 030332 012737 030340 001110 MOV #1$, $LPERR ;:LOAD LOOP ON ERROR LOCATION FOR
4986 ; SUBTEST LOOP
4987
4988 030340 1$: CLR RKWC(R2) ;:CLEAR WORD COUNT REG.
4989 030340 005062 000002 MOV CONFIG, RKCS2(R2) ;:WRITE RKCS2
4990 030344 013762 002010 000010 MOV RKCS2(R2), $BDDAT ;:STORE RKCS2
4991 030352 016237 000010 001126 MOV CONFIG, $GDDAT ;:PREPARE EXPECTED RESULTS
4992 030360 013737 002010 001124 MOV #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR, $GDDAT
4993 030366 042737 177600 001124 BIC
    
```



D 8

CZR6AD0 RK611 DSKLS CTRL PRT1 MACY11 30(1046) 14-SEP-81 15:04 PAGE 95  
 CZR6AD.P11 14-SEP-81 13:43 T36 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 2) SEQ 0094

4994	030374	052737	000100	001124	BIS	#IR,\$GDDAT	:INITIALIZE READ ONLY BITS
4995	030402	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKCS2 CORRECT
4996	030410	001404			BEQ	2\$	:YES,TEST IF ANY OTHER REG MODIFIED
4997	030412	012737	066766	001322	MOV	#EM1004,EM3N+2	:LOAD ERROR MESSAGE
4998	030420	104003			ERROR	3	
4999	030422						
5000	030422	016237	000000	001126	2\$:	MOV	RKCS1(R2),\$BDDAT :STORE COMMAND AND STATUS REG. 1
5001	030430	022737	000200	001126		CMP	#RDY,\$BDDAT :CHECK IF CS1 CORRECT
5002	030436	001407				BEQ	3\$ :YES, CONTINUE
5003	030440	012737	000100	001124		MOV	#IR,\$GDDAT :LOAD EXPECTED RESULTS
5004	030446	012737	067376	001322		MOV	#EM1017,EM3N+2 :LOAD ERROR MESSAGE
5005	030454	104003				ERROR	3
5006	030456						
5007	030456	016237	000004	001126	3\$:	MOV	RKBA(R2),\$BDDAT :STORE BUS AND REG
5008	030464	001406				BEQ	4\$ :CHECK IF ZERO
5009	030466	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5010	030472	012737	067447	001322		MOV	#EM1019,EM3N+2 :LOAD ERROR MESSAGE
5011	030500	104003				ERROR	3
5012	030502						
5013	030502	016237	000002	001126	4\$:	MOV	RKWC(R2),\$BDDAT :STORE WORD COUNT REG
5014	030510	001406				BEQ	5\$ :CHECK IF ZERO
5015	030512	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5016	030516	012737	067417	001322		MOV	#EM1018,EM3N+2 :LOAD ERROR MESSAGE
5017	030524	104003				ERROR	3
5018	030526						
5019	030526	016237	000006	001126	5\$:	MOV	RKDA(R2),\$BDDAT :STORE DISK AVERAGE REG
5020	030534	001406				BEQ	6\$ :CHECK IF ZERO
5021	030536	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5022	030542	012737	067474	001322		MOV	#EM1020,EM3N+2 :LOAD ERROR MESSAGE
5023	030550	104003				ERROR	3
5024	030552						
5025	030552	016237	000016	001126	6\$:	MOV	RKASOF(R2),\$BDDAT :STORE ATTENTION SUMMARY/OFFSET REG.
5026	030560	001406				BEQ	7\$ :CHECK IF ZERO
5027	030562	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5028	030566	012737	067640	001322		MOV	#EM1024,EM3N+2 :LOAD ERROR MESSAGE
5029	030574	104003				ERROR	3
5030	030576						
5031	030576	016237	000012	001126	7\$:	MOV	RKDS(R2),\$BDDAT :STORE DRIVE STATUS REG
5032	030604	001406				BEQ	9\$ :CHECK IF ZERO
5033	030606	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5034	030612	012737	067553	001322		MOV	#EM1022,EM3N+2 :LOAD ERROR MESSAGE
5035	030620	104003				ERROR	3
5036	030622	016237	000014	001126	9\$:	MOV	RKER(R2),\$BDDAT :STORE ERROR REG
5037	030630	001406				BEQ	10\$ :CHECK IF ZERO
5038	030632	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5039	030636	012737	067611	001322		MOV	#EM1023,EM3N+2 :LOAD ERROR MESSAGE
5040	030644	104003				ERROR	3
5041	030646						
5042	030646	016237	000020	001126	10\$:	MOV	RKDCYL(R2),\$BDDAT :STORE CYLINDER ADD REG
5043	030654	001406				BEQ	12\$ :CHECK IF ZERO
5044	030656	005037	001124			CLR	\$GDDAT :LOAD EXPECTED CONTENTS
5045	030662	012737	067706	001322		MOV	#EM1025,EM3N+2 :LOAD ERROR MESSAGE
5046	030670	104003				ERROR	3
5047	030672						
5048	030672	016237	000026	001126	12\$:	MOV	RKMR1(R2),\$BDDAT :STORE MAINTENANCE REG.1
5049	030700	012737	002000	001124		MOV	#MEWD,\$GDDAT :LOAD EXPECTED MR1



```

5050 030706 032737 020000 001126 BIT #ECCW,$BDDAT
5051 030714 001403 BEQ 13$
5052 030716 052737 020000 001124 BIS #ECCW,$GDDAT
5053 030724 023737 001124 001126 13$: CMP $GDDAT,$BDDAT :CHECK IF MR1 CORRECT
5054 030732 001404 BEQ 14$ :YES,ISSUE CONTROLLER CLEAR
5055 030734 012737 067733 001322 MOV #EM1026,EM3N+2 :LOAD ERROR MESSAGE
5056 030742 104003 ERROR 3
5057 030744 14$:
5058 030744 016237 000032 001126 MOV RKECPT(R2),$BDDAT :STORE ECC PATTERN REG.
5059 030752 001406 BEQ 15$ :CHECK IF ZERO
5060 030754 005037 001124 CLR $GDDAT :LOAD EXPECTED CONTENTS
5061 030760 012737 070011 001322 MOV #EM1030,EM3N+2 :LOAD ERROR MESSAGE
5062 030766 104003 ERROR 3
5063 030770 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT :STORE ECC POSITION REG.
5064 030776 012737 004066 001124 16$: MOV #4066,$GDDAT :USE 4066
5065 031004 023737 001124 001126 17$: CMP $GDDAT,$BDDAT :CHECK IF ECC POSITION CORRECT
5066 031012 001404 BEQ 18$ :YES,INITIALIZE RK611
5067 031014 012737 067764 001322 MOV #EM1029,EM3N+2 :LOAD ERROR MESSAGE
5068 031022 104003 ERROR 3
5069 031024 016237 000010 002014 18$: MOV RKCS2(R2),PRREG :GET PREVIOUS CONTENTS
5070 031032 012762 100000 000000 MOV #CCLR,RKCS1(R2) :CLEAR RK611 CONTROLLER
5071 031040 016237 000010 001126 MOV RKCS2(R2),$BDDAT :GET CURRENT VALUE
5072 031046 012737 000100 001124 MOV #IR,$GDDAT :LOAD EXPECTED CONTENTS
5073 031054 023737 001124 001126 CMP $GDDAT,$BDDAT :CHECK IF RKCS2 CORRECT
5074 031062 001407 BEQ 19$ :YES, CHECK IF FINISHED
5075 031064 012737 063465 001330 MOV #EM3,EM4N :LOAD ERROR MESSAGE
5076 031072 012737 066766 001332 MOV #EM1004,EM4N+2
5077 031100 104004 ERROR 4
5078 031102 104415 19$: SCOP1 :CHECK IF LOOP ON ERROR
5079 031104 000261 SEC :SHIFT IN ONE
5080 031106 006137 002012 ROL CONFIG1
5081 031112 013737 002012 002010 MOV CONFIG1,CONFIG :MAKE SURE SUBSYSTEM CLEAR
5082 031120 042737 000040 002010 BIC #SCLR,CONFIG : DOES NOT SET
5083 031126 005301 DEC R1 :CHECK IF FINISHED
5084 031130 001402 BEQ TST37 :YES, GO ON TO NEXT TEST
5085 031132 000137 030340 JMP 1$
    
```

```

*****
*TEST 37 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 3)
*
* RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
* WRITE THE WORD COUNT REGISTER WITH ZERO.
*
* WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS
* SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS
* REG. 2 AND CHECK FOR REGISTER INTERACTION.
*
* 000001 000037 001737 037737
* 000003 000137 003737 077737
* 000007 000337 007737 177737
* 000017 000737 017737 000000
    
```

```

5103 031136 000004 TST37: SCOPE
5104 031140 012737 000764 001200 MOV #500,$TIMES ;;DO 500. ITERATIONS
5105 031146 012701 000021 MOV #17.,R1 :LOAD NUMBER OF PATTERNS
    
```



5106	031152	005037	002010		CLR	CONFIG	:LOAD INITIAL OCNFIGURATION
5107	031156	005037	002012		CLR	CONFIG1	
5108	031162	012737	064510	001320	MOV	#EM18,EM3N	:LOAD ERROR MESSAGE
5109	031170	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	:CLEAR RK611 WITH CONTROLLER CLEAR
5110	031176	012737	031204	001110	MOV	#1\$, \$LPERR	:LOAD LOOP ON ERROR LOCATION FOR
5111							: SUBTEST LOOP
5112							
5113	031204				1\$:		
5114	031204	005062	000002		CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
5115	031210	013762	002010	000010	MOV	CONFIG,RKCS2(R2)	:WRITE RKCS2
5116	031216	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE RKCS2
5117	031224	013737	002010	001124	MOV	CONFIG, \$GDDAT	:PREPARE EXPECTED RESULTS
5118	031232	042737	177600	001124	BIC	#DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR, \$GDDAT	
5119	031240	052737	000100	001124	BIS	#IR, \$GDDAT	:INITIALIZE READ ONLY BITS
5120	031246	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF RKCS2 CORRECT
5121	031254	001404			BEQ	2\$	:YES, TEST IF ANY OTHER REG MODIFIED
5122	031256	012737	066766	001322	MOV	#EM1004, EM3N+2	:LOAD ERROR MESSAGE
5123	031264	104003			ERROR	3	
5124	031266				2\$:		
5125	031266	016237	000000	001126	MOV	RKCS1(R2), \$BDDAT	:STORE COMMAND AND STATUS REG. 1
5126	031274	022737	000200	001126	CMP	#RDY, \$BDDAT	:CHECK IF CS1 CORRECT
5127	031302	001407			BEQ	3\$	:YES, CONTINUE
5128	031304	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED RESULTS
5129	031312	012737	067376	001322	MOV	#EM1017, EM3N+2	:LOAD ERROR MESSAGE
5130	031320	104003			ERROR	3	
5131	031322				3\$:		
5132	031322	016237	000004	001126	MOV	RKBA(R2), \$BDDAT	:STORE BUS AND REG
5133	031330	001406			BEQ	4\$	:CHECK IF ZERO
5134	031332	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5135	031336	012737	067447	001322	MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
5136	031344	104003			ERROR	3	
5137	031346				4\$:		
5138	031346	016237	000002	001126	MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG
5139	031354	001406			BEQ	5\$	:CHECK IF ZERO
5140	031356	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5141	031362	012737	067417	001322	MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
5142	031370	104003			ERROR	3	
5143	031372				5\$:		
5144	031372	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
5145	031400	001406			BEQ	6\$	:CHECK IF ZERO
5146	031402	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5147	031406	012737	067474	001322	MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
5148	031414	104003			ERROR	3	
5149	031416				6\$:		
5150	031416	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
5151	031424	001406			BEQ	7\$	:CHECK IF ZERO
5152	031426	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5153	031432	012737	067640	001322	MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
5154	031440	104003			ERROR	3	
5155	031442				7\$:		
5156	031442	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
5157	031450	001406			BEQ	9\$	:CHECK IF ZERO
5158	031452	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5159	031456	012737	067553	001322	MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
5160	031464	104003			ERROR	3	
5161	031466	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT :STORE ERROR REG



5162	031474	001406				BEQ	10\$		:CHECK IF ZERO
5163	031476	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
5164	031502	012737	067611	001322		MOV	#EM1023,EM3N+2		:LOAD ERROR MESSAGE
5165	031510	104003				ERROR	3		
5166	031512				10\$:				
5167	031512	016237	000020	001126		MOV	RKDCYL(R2),\$BDDAT		:STORE CYLINDER ADD REG
5168	031520	001406				BEQ	12\$		:CHECK IF ZERO
5169	031522	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
5170	031526	012737	067706	001322		MOV	#EM1025,EM3N+2		:LOAD ERROR MESSAGE
5171	031534	104003				ERROR	3		
5172	031536				12\$:				
5173	031536	016237	000026	001126		MOV	RKMR1(R2),\$BDDAT		:STORE MAINTENANCE REG.1
5174	031544	012737	002000	001124		MOV	#MEWD,\$GDDAT		:LOAD EXPECTED MR1
5175	031552	032737	020000	001126		BIT	#ECCW,\$BDDAT		
5176	031560	001403				BEQ	13\$		
5177	031562	052737	020000	001124		BIS	#ECCW,\$GDDAT		
5178	031570	023737	001124	001126	13\$:	CMP	\$GDDAT,\$BDDAT		:CHECK IF MR1 CORRECT
5179	031576	001404				BEQ	14\$		:YES,ISSUE CONTROLLER CLEAR
5180	031600	012737	067733	001322		MOV	#EM1026,EM3N+2		:LOAD ERROR MESSAGE
5181	031606	104003				ERROR	3		
5182	031610				14\$:				
5183	031610	016237	000032	001126		MOV	RKECPT(R2),\$BDDAT		:STORE ECC PATTERN REG.
5184	031616	001406				BEQ	15\$		:CHECK IF ZERO
5185	031620	005037	001124			CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
5186	031624	012737	070011	001322		MOV	#EM1030,EM3N+2		:LOAD ERROR MESSAGE
5187	031632	104003				ERROR	3		
5188	031634	016237	000030	001126	15\$:	MOV	RKECPS(R2),\$BDDAT		:STORE ECC POSITION REG.
5189	031642	012737	004066	001124	16\$:	MOV	#4066,\$GDDAT		:USE 4066
5190	031650	023737	001124	001126	17\$:	CMP	\$GDDAT,\$BDDAT		:CHECK IF ECC POSITION CORRECT
5191	031656	001404				BEQ	18\$		:YES,INITIALIZE RK611
5192	031660	012737	067764	001322		MOV	#EM1029,EM3N+2		:LOAD ERROR MESSAGE
5193	031666	104003				ERROR	3		
5194	031670	016237	000010	002014	18\$:	MOV	RKCS2(R2),PREREG		:GET PREVIOUS CONTENTS
5195	031676	012762	100000	000000		MOV	#CCLR,RKCS1(R2)		:CLEAR RK611 CONTROLLER
5196	031704	016237	000010	001126		MOV	RKCS2(R2),\$BDDAT		:GET CURRENT VALUE
5197	031712	012737	000100	001124		MOV	#IR,\$GDDAT		:LOAD EXPECTED CONTENTS
5198	031720	023737	001124	001126		CMP	\$GDDAT,\$BDDAT		:CHECK IF RKCS2 CORRECT
5199	031726	001407				BEQ	19\$		:YES, CHECK IF FINISHED
5200	031730	012737	063465	001330		MOV	#EM3,EM4N		:LOAD ERROR MESSAGE
5201	031736	012737	066766	001332		MOV	#EM1004,EM4N+2		
5202	031744	104004				ERROR	4		
5203	031746	104415			19\$:	SCOP1			:CHECK IF LOOP ON ERROR
5204	031750	000261				SEC			:SHIFT IN ONE
5205	031752	006137	002012			ROL	CONFIG1		
5206	031756	013737	002012	002010		MOV	CONFIG1,CONFIG		:MAKE SURE SUBSYSTEM CLEAR
5207	031764	042737	000040	002010		BIC	#SCLR,CONFIG		: DOES NOT SET
5208	031772	005301				DEC	R1		:CHECK IF FINISHED
5209	031774	001402				BEQ	TST40		:;YES, GO ON TO NEXT TEST
5210	031776	000137	031204			JMP	1\$		

```

*****
*TEST 40 REGISTER INTERACTION USING COMMAND STATUS REG. 2 (PART 4)
*
*
* RESET THE RK611 REGISTERS WITH A CONTROLLER CLEAR.
* WRITE THE WORD COUNT REGISTER WITH ZERO.
*
*****

```

5211  
5212  
5213  
5214  
5215  
5216  
5217



5218  
5219  
5220  
5221  
5222  
5223  
5224  
5225  
5226  
5227  
5228 032002 000004  
5229 032004 012737 000764 001200  
5230 032012 012701 000021  
5231 032016 005037 002010  
5232 032022 005037 002012  
5233 032026 012737 064510 001320  
5234 032034 012762 100000 000000  
5235 032042 012737 032050 001110  
5236  
5237  
5238 032050  
5239 032050 005062 000002  
5240 032054 013762 002010 000010  
5241 032062 016237 000010 001126  
5242 032070 013737 002010 001124  
5243 032076 042737 177600 001124  
5244 032104 052737 000100 001124  
5245 032112 023737 001124 001126  
5246 032120 001404  
5247 032122 012737 066766 001322  
5248 032130 104003  
5249 032132  
5250 032132 016237 000000 001126  
5251 032140 022737 000200 001126  
5252 032146 001407  
5253 032150 012737 000100 001124  
5254 032156 012737 067376 001322  
5255 032164 104003  
5256 032166  
5257 032166 016237 000004 001126  
5258 032174 001406  
5259 032176 005037 001124  
5260 032202 012737 067447 001322  
5261 032210 104003  
5262 032212  
5263 032212 016237 000002 001126  
5264 032220 001406  
5265 032222 005037 001124  
5266 032226 012737 067417 001322  
5267 032234 104003  
5268 032236  
5269 032236 016237 000006 001126  
5270 032244 001406  
5271 032246 005037 001124  
5272 032252 012737 067474 001322  
5273 032260 104003

WRITE COMMAND AND STATUS REGISTER 2 WITH THE FOLLOWING CONFIGURATIONS  
SUCH THAT SUBSYSTEM CLEAR IS RESET. CHECK COMMAND AND STATUS  
REG. 2 AND CHECK FOR REGISTER INTERACTION.

100000 174000 177600 177734  
140000 176000 177700 177736  
160000 177000 177720 177737  
170000 177400 177730 000000

\*\*\*\*\*

TST40: SCOPE  
MOV #500.,\$TIMES ;:DO 500. ITERATIONS  
MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS  
CLR CONFIG ;:LOAD INITIAL CCONFIGURATION  
CLR CONFG1  
MOV #EM18,EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG,RKCS2(R2) ;:WRITE RKCS2  
MOV RKCS2(R2), \$BDDAT ;:STORE RKCS2  
MOV CONFIG,\$GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #DLT!WCE!UPE!NED!NEM!PGE!MDS!UFE!OR,\$GDDAT  
BIS #IR,\$GDDAT ;:INITIALIZE READ ONLY BITS  
CMP \$GDDAT,\$BDDAT ;:CHECK IF RKCS2 CORRECT  
BEQ 2\$ ;:YES,TEST IF ANY OTHER REG MODIFIED  
MOV #EM1004,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

2\$:  
MOV RKCS1(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY,\$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3\$ ;:YES, CONTINUE  
MOV #IR,\$GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

3\$:  
MOV RKBA(R2), \$BDDAT ;:STORE BUS AND REG  
BEQ 4\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

4\$:  
MOV RKWC(R2), \$BDDAT ;:STORE WORD COUNT REG  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3

5\$:  
MOV RKDA(R2), \$BDDAT ;:STORE DISK AVERAGE REG  
BEQ 6\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1020,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3



5274	032262				6\$:		
5275	032262	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
5276	032270	001406				BEQ	7\$ ;CHECK IF ZERO
5277	032272	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
5278	032276	012737	067640	001322		MOV	#EM1024, EM3N+2 ;LOAD ERROR MESSAGE
5279	032304	104003				ERROR	3
5280	032306				7\$:		
5281	032306	016237	000012	001126		MOV	RKDS(R2), \$BDDAT ;STORE DRIVE STATUS REG
5282	032314	001406				BEQ	9\$ ;CHECK IF ZERO
5283	032316	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
5284	032322	012737	067553	001322		MOV	#EM1022, EM3N+2 ;LOAD ERROR MESSAGE
5285	032330	104003				ERROR	3
5286	032332	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT ;STORE ERROR REG
5287	032340	001406				BEQ	10\$ ;CHECK IF ZERO
5288	032342	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
5289	032346	012737	067611	001322		MOV	#EM1023, EM3N+2 ;LOAD ERROR MESSAGE
5290	032354	104003				ERROR	3
5291	032356				10\$:		
5292	032356	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT ;STORE CYLINDER ADD REG
5293	032364	001406				BEQ	12\$ ;CHECK IF ZERO
5294	032366	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
5295	032372	012737	067706	001322		MOV	#EM1025, EM3N+2 ;LOAD ERROR MESSAGE
5296	032400	104003				ERROR	3
5297	032402				12\$:		
5298	032402	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT ;STORE MAINTENANCE REG.1
5299	032410	012737	002000	001124		MOV	#MEWD, \$GDDAT ;LOAD EXPECTED MP1
5300	032416	032737	020000	001126		BIT	#ECCW, \$BDDAT
5301	032424	001403				BEQ	13\$
5302	032426	052737	020000	001124		BIS	#ECCW, \$GDDAT
5303	032434	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT ;CHECK IF MR1 CORRECT
5304	032442	001404				BEQ	14\$ ;YES, ISSUE CONTROLLER CLEAR
5305	032444	012737	067733	001322		MOV	#EM1026, EM3N+2 ;LOAD ERROR MESSAGE
5306	032452	104003				ERROR	3
5307	032454				14\$:		
5308	032454	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT ;STORE ECC PATTERN REG.
5309	032462	001406				BEQ	15\$ ;CHECK IF ZERO
5310	032464	005037	001124			CLR	\$GDDAT ;LOAD EXPECTED CONTENTS
5311	032470	012737	070011	001322		MOV	#EM1030, EM3N+2 ;LOAD ERROR MESSAGE
5312	032476	104003				ERROR	3
5313	032500	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT ;STORE ECC POSITION REG.
5314	032506	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT ;USE 4066
5315	032514	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT ;CHECK IF ECC POSITION CORRECT
5316	032522	001404				BEQ	18\$ ;YES, INITIALIZE RK611
5317	032524	012737	067764	001322		MOV	#EM1029, EM3N+2 ;LOAD ERROR MESSAGE
5318	032532	104003				ERROR	3
5319	032534	016237	000010	002014	18\$:	MOV	RKCS2(R2), PREREG ;GET PREVIOUS CONTENTS
5320	032542	012762	100000	000000		MOV	#CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
5321	032550	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT ;GET CURRENT VALUE
5322	032556	012737	00C100	001124		MOV	#IR, \$GDDAT ;LOAD EXPECTED CONTENTS
5323	032564	023737	001124	001126		CMP	\$GDDAT, \$BDDAT ;CHECK IF RKCS2 CORRECT
5324	032572	001407				BEQ	19\$ ;YES, CHECK IF FINISHED
5325	032574	012737	063465	001330		MOV	#EM3, EM4N ;LOAD ERROR MESSAGE
5326	032602	012737	066766	001332		MOV	#EM1004, EM4N+2
5327	032610	104004				ERROR	4
5328	032612	104415			19\$:	SCOP1	;CHECK IF LOOP ON ERROR
5329	032614	000261				SEC	;SHIFT IN ONE



```

5330 032616 006037 002012 ROR CONFIG
5331 032622 013737 002012 002010 MOV CONFIG,CONFIG ;MAKE SURE SUBSYSTEM CLEAR
5332 032630 042737 000040 002010 BIC #SCLR,CONFIG ; DOES NOT SET
5333 032636 005301 DEC R1 ;CHECK IF FINISHED
5334 032640 001402 BEQ TST41 ;:YES, GO ON TO NEXT TEST
5335 032642 000137 032050 JMP 1$

*****
:TEST 41 CHECK SUBSYSTEM CLEAR WITH BUS ADDRESS
*****
:
: THIS TEST WILL TEST THE ABILITY OF THE SUBSYSTEM CLEAR TO
: INITIALIZE THE BUS ADDRESS REGISTER AND COMMAND
: AND STATUS REGISTER 1. IT WILL ALSO VERIFY THAT ALL
: OTHER REGISTERS REMAIN IN THE INITIALIZED STATE.
*****
TST41: SCOPE
5346 032646 000004 MOV #500, $TIMES ;:DO 500. ITERATIONS
5347 032650 012737 000764 001200 MOV $BASE,R2 ;:LOAD RK611 BASE
5348 032656 013702 001270 MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR
5349 032662 012762 100000 000000 CLR RKWC(R2) ;:CLEAR WORD COUNT REG
5350 032670 005062 000002 MOV #177776,RKBA(R2) ;:WRITE WORD COUNT REG.
5351 032674 012762 177776 000004 MOV #013776,RKCS1(R2) ;:WRITE COMMAND AND STATUS REG 1
5352 032702 012762 013776 000000 MOV #SCLR,RKCS2(R2) ;:ISSUE A SUBSYSTEM CLEAR
5353 032710 012762 000040 000010 MOV RKBA(R2), $BDDAT ;:STORE BUS ADDRESS REG.
5354 032716 016237 000004 001126 BEQ 1$ ;:CHECK IF ZERO
5355 032724 001414 MOV #177776,PREREG ;:LOAD PREVIOUS CONTENTS
5356 032726 012737 177776 002014 CLR $GDDAT ;:LOAD EXPECTED CONTENTS
5357 032734 005037 001124 MOV #EM21,EM4N ;:LOAD ERROR MESSAGE
5358 032740 012737 064641 001330 MOV #EM1002,EM4N
5359 032746 012737 066721 001330 ERROR 4
5360 032754 104004 1$: MOV RKCS1(R2), $BDDAT ;:STORE COMMAND AND STATUS REG.1
5361 032756 016237 000000 001126 CMP #RDY,$BDDAT ;:CHECK IF CS1 CORRECT
5362 032764 022737 000200 001126 BEQ 2$ ;:YES, CHECK IF ALL OTHER REGISTER
5363 032772 001415 ;: IN INITIALIZE STATE
5364 MOV #013776,PREREG ;:LOAD PREVIOUS CONTENTS
5365 032774 012737 013776 002014 MOV #RDY,$GDDAT ;:LOAD EXPECTED CONTENTS
5366 033002 012737 000200 001124 MOV #EM22,EM4N; ;:LOAD ERROR MESSAGE
5367 033010 012737 064716 001330 MOV #EM1002,EM4N+2
5368 033016 012737 066721 001332 ERROR 4
5369 033024 104004 2$: MOV #EM23,EM2N ;:LOAD ERROR MESSAGE
5370 033026 012737 064767 001310 MOV RKWC(R2), $BDDAT ;:STORE WORD COUNT REG.
5371 033034 016237 000002 001126 BEQ 3$ ;:CHECK IF ZERO
5372 033042 001406 CLR $GDDAT ;:LOAD EXPECTED CONTENTS
5373 033044 005037 001124 MOV #EM1001,EM2N+2 ;:LOAD ERROR MESSAGE
5374 033050 012737 066674 001312 ERROR 2
5375 033056 104002 3$: MOV RKDA(R2), $BDDAT ;:STORE DISK ADDRESS REG.
5376 033060 016237 000006 001126 BEQ 4$ ;:CHECK IF ZERO
5377 033066 001406 CLR $GDDAT ;:LOAD EXPECTED CONTENTS
5378 033070 005037 001124 MOV #EM1003,EM2N+2 ;:LOAD ERROR MESSAGE
5379 033074 012737 066743 001312 ERROR 2
5380 033102 104002 4$: MOV RKASOF(R2), $BDDAT ;:STORE ATTENTION SUMMARY AND OFFSET REG.
5381 033104 016237 000016 001126 BEQ 5$ ;:CHECK IF ZERO
5382 033112 001406 CLR $GDDAT ;:LOAD EXPECTED CONTENTS
5383 033114 005037 001124 MOV #EM1007,EM2N+2 ;:LOAD ERROR MESSAGE
5384 033120 012737 067063 001312 ERROR 2
5385 033126 104002
    
```



```

5386 033130 016237 000010 001126 5$: MOV RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG.2
5387 033136 022737 000100 001126 CMP #IR, $BDDAT ;CHECK IF CS2 CORRECT
5388 033144 001407 BEQ 6$ ;YES, CONTINUE
5389 033146 012737 000100 001124 MOV #IR, $GDDAT ;LOAD EXPECTED CONTENTS
5390 033154 012737 066766 001312 MOV #EM1004, EM2N+2 ;LOAD ERROR MESSAGE
5391 033162 104002 ERROR 2
5392 033164 016237 000012 001126 6$: MOV RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG.
5393 033172 001406 BEQ 7$ ;CHECK IF ZERO
5394 033174 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5395 033200 012737 067004 001312 MOV #EM1005, EM2N+2 ;LOAD ERROR MESSAGE
5396 033206 104002 ERROR 2
5397 033210 016237 000014 001126 7$: MOV RKER(R2), $BDDAT ;STORE ERROR REGISTER
5398 033216 001406 BEQ 8$ ;CHECK IF ZERO
5399 033220 005037 001126 CLR $BDDAT ;LOAD EXPECTED CONTENTS
5400 033224 012737 067037 001312 MOV #EM1006, EM2N+2 ;LOAD ERROR MESSAGE
5401 033232 104002 ERROR 2
5402 033234 016237 000020 001126 8$: MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG.
5403 033242 001406 BEQ 10$ ;CHECK IF ZERO
5404 033244 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5405 033250 012737 067354 001312 MOV #EM1016, EM2N+2 ;LOAD ERROR MESSAGE
5406 033256 104002 ERROR 2
5407 033260 016237 000026 001126 10$: MOV RKMRI(R2), $BDDAT ;STORE CYLINDER ADD REG.
5408 033266 012737 002000 001124 MOV #MEWD, $GDDAT ;LOAD EXPECTED MR1
5409 033274 032737 020000 001126 BIT #ECCW, $BDDAT
5410 033302 001403 BEQ 11$
5411 033304 052737 020000 001124 BIS #ECCW, $GDDAT
5412 033312 023737 001124 001126 11$: CMP $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
5413 033320 001404 BEQ 12$ ;YES, CONTINUE TEST
5414 033322 012737 067154 001312 MOV #EM1009, EM2N+2 ;LOAD ERROR MESSAGE
5415 033330 104002 ERROR 2
5416 033332 016237 000032 001126 12$: MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
5417 033340 001406 BEQ 13$ ;CHECK IF ZERO
5418 033342 005037 001124 CLR $GDDAT ;LOAD EXPECTED RESULTS
5419 033346 012737 067224 001312 MOV #EM1013, EM2N+2 ;LOAD ERROR MESSAGE
5420 033354 104002 ERROR 2
5421 033356 016237 000030 001126 13$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
5422 033364 022737 004066 001126 CMP #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
5423 033372 001407 BEQ TST42 ;:YES, GO TO NEXT TEST
5424 033374 012737 004066 001124 MOV #4066, $GDDAT ;LOAD EXPECTED RESULTS
5425 033402 012737 067202 001312 MOV #EM1012, EM2N+2 ;LOAD ERROR MESSAGE
5426 033410 104002 ERROR 2
5427 *****
5428 :*TEST 42 REGISTER INTERACTION USING DRIVE STATUS
5429 :*
5430 :* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5431 :* WRITE WORD COUNT TO 0, WRITE DRIVE STATUS REGISTER
5432 :* WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
5433 :* INTERACTION TAKES PLACE.
5434 :*
5435 *****
5436 033412 000004 TST42: SCOPE
5437 033414 012737 000764 001200 MOV #500, $TIMES ;:DO 500. ITERATIONS
5438 033422 013702 001270 MOV $BASE, R2 ;LOAD RK611 BASE ADDRESS
5439 033426 012737 177777 002010 MOV #177777, CONFIG ;LOAD CONFIGURATION WORD
5440 033434 012737 064077 001320 MOV #EM10, EM3N ;LOAD ERROR MESSAGE
5441 033442 012762 100000 000000 MOV #CLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
    
```



5442	033450	005062	000002			CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
5443	033454	012762	177777	000012		MOV	#177777,RKDS(R2)	:WRITE RKDS WITH 177777
5444	033462	016237	000000	001126		MOV	RKCS1(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.1
5445	033470	022737	000200	001126		CMP	#RDY, \$BDDAT	:CHECK IF CS1 CORRECT
5446	033476	001407				BEQ	1\$	:YES, CONTINUE
5447	033500	012737	000200	001124		MOV	#RDY, \$GDDAT	:LOAD EXPECTED RESULTS
5448	033506	012737	067376	001322		MOV	#EM1017, EM3N+2	:LOAD ERROR MESSAGE
5449	033514	104003				ERROR	3	
5450	033516	016237	000004	001126	1\$:	MOV	RKBA(R2), \$BDDAT	:STORE BUS ADD REG.
5451	033524	001406				BEQ	2\$	:CHECK IF ZERO
5452	033526	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5453	033532	012737	067447	001322		MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
5454	033540	104003				ERROR	3	
5455	033542	016237	000002	001126	2\$:	MOV	RKWC(R2), \$BDDAT	:STORE WORK COUNT REG.
5456	033550	001406				BEQ	3\$	:CHECK IF ZERO
5457	033552	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5458	033556	012737	067417	001322		MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
5459	033564	104003				ERROR	3	
5460	033566	016237	000006	001126	3\$:	MOV	RKDA(R2), \$BDDAT	:STORE DISK ADD REG
5461	033574	001406				BEQ	4\$	:CHECK IF ZERO
5462	033576	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5463	033602	012737	067474	001322		MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
5464	033610	104003				ERROR	3	
5465	033612	016237	000016	001126	4\$:	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG
5466	033620	001406				BEQ	5\$	:CHECK IF ZERO
5467	033622	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5468	033626	012737	067640	001322		MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
5469	033634	104003				ERROR	3	
5470	033636	016237	000010	001126	5\$:	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG. 2
5471	033644	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
5472	033652	001407				BEQ	6\$	:YES, CONTINUE
5473	033654	012737	000100	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
5474	033662	012737	067532	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
5475	033670	104003				ERROR	3	
5476	033672	016237	000012	001126	6\$:	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG.
5477	033700	001406				BEQ	7\$	:CHECK IF ZERO
5478	033702	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5479	033706	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
5480	033714	104003				ERROR	3	
5481	033716	016237	000014	001126	7\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
5482	033724	001406				BEQ	8\$	:CHECK IF ZERO
5483	033726	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5484	033732	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
5485	033740	104003				ERROR	3	
5486	033742	016237	000020	001126	8\$:	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
5487	033750	001406				BEQ	10\$	:CHECK IF ZERO
5488	033752	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED RESULTS
5489	033756	012737	067706	001322		MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
5490	033764	104003				ERROR	3	
5491	033766	016237	000026	001126	10\$:	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG 1
5492	033774	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED CONTENTS
5493	034002	032737	020000	001126		BIT	#ECCW, \$BDDAT	
5494	034010	001403				BEQ	11\$	
5495	034012	052737	020000	001124		BIS	#ECCW, \$GDDAT	
5496	034020	023737	001124	001126	11\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
5497	034026	001404				BEQ	12\$	:YES, CONTINUE TEST



```

5498 034030 012737 067733 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
5499 034036 104003      ERROR    3
5500 034040 016237 000032 001126 12$:  MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
5501 034046 001406      BEQ      13$ ;CHECK IF ZERO
5502 034050 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5503 034054 012737 070011 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5504 034062 104003      ERROR    3
5505 034064 016237 000030 001126 13$:  MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG:
5506 034072 022737 004066 001126      CMP      #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
5507 034100 001407      BEQ      TST43 ;:YES, GO TO NEXT TEST
5508 034102 012737 004066 001124      MOV      #4066, $GDDAT ;LOAD EXPECTED CONTENTS
5509 034110 012737 067764 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5510 034116 104003      ERROR    3
5511
5512
5513      :*****
5514      :*TEST 43 REGISTER INTERACTION USING ERROR REGISTER
5515      :*
5516      :* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5517      :* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE ERROR REGISTER
5518      :* WITH 177777 AND MAKE SURE IT REMAINS 0 AND NO
5519      :* INTERACTION TAKES PLACE.
5520      :*****
5521 034120 000004      TST43:  SCOPE
5522 034122 012737 000764 001200      MOV      #500, $TIMES ;:DO 500. ITERATIONS
5523 034130 013702 001270      MOV      $BASE, R2 ;LOAD RK611 BASE ADDRESS
5524 034134 012737 177777 002010      MOV      #177777, CONFIG ;LOAD CONFIGURATION WORD
5525 034142 012737 064144 001320      MOV      #EM11, EM3N ;LOAD ERROR MESSAGE
5526 034150 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5527 034156 005062 000002      CLR      RKWC(R2) ;CLEAR WORD COUNT REG.
5528 034162 012762 177777 000014      MOV      #177777, RKER(R2) ;WRITE RKER WITH 177777
5529 034170 016237 000000 001126      MOV      RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG.1
5530 034176 022737 000200 001126      CMP      #RDY, $BDDAT ;CHECK IF CS1 CORRECT
5531 034204 001407      BEQ      1$ ;YES, CONTINUE
5532 034206 012737 000200 001124      MOV      #RDY, $GDDAT ;LOAD EXPECTED RESULTS
5533 034214 012737 067376 001322      MOV      #EM1017, EM3N+2 ;LOAD ERROR MESSAGE
5534 034222 104003      ERROR    3
5535 034224 016237 000004 001126 1$:  MOV      RKBA(R2), $BDDAT ;STORE BUS ADD REG.
5536 034232 001406      BEQ      2$ ;CHECK IF ZERO
5537 034234 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5538 034240 012737 067447 001322      MOV      #EM1019, EM3N+2 ;LOAD ERROR MESSAGE
5539 034246 104003      ERROR    3
5540 034250 016237 000002 001126 2$:  MOV      RKWC(R2), $BDDAT ;STORE WORK COUNT REG.
5541 034256 001406      BEQ      3$ ;CHECK IF ZERO
5542 034260 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5543 034264 012737 067417 001322      MOV      #EM1018, EM3N+2 ;LOAD ERROR MESSAGE
5544 034272 104003      ERROR    3
5545 034274 016237 000006 001126 3$:  MOV      RKDA(R2), $BDDAT ;STORE DISK ADD REG
5546 034302 001406      BEQ      4$ ;CHECK IF ZERO
5547 034304 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5548 034310 012737 067474 001322      MOV      #EM1020, EM3N+2 ;LOAD ERROR MESSAGE
5549 034316 104003      ERROR    3
5550 034320 016237 000016 001126 4$:  MOV      RKASOF(R2), $BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
5551 034326 001406      BEQ      5$ ;CHECK IF ZERO
5552 034330 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5553 034334 012737 067640 001322      MOV      #EM1024, EM3N+2 ;LOAD ERROR MESSAGE
    
```



```

5554 034342 104003          ERROR 3
5555 034344 016237 000010 001126 5$: MOV RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 2
5556 034352 022737 000100 001126    CMP #IR, $BDDAT ;CHECK IF CS2 CORRECT
5557 034360 001407          BEQ 6$ ;YES, CONTINUE
5558 034362 012737 000100 001124    MOV #IR, $GDDAT ;LOAD EXPECTED CONTENTS
5559 034370 012737 067532 001322    MOV #EM1021, EM3N+2 ;LOAD ERROR MESSAGE
5560 034376 104003          ERROR 3
5561 034400 016237 000012 001126 6$: MOV RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG.
5562 034406 001406          BEQ 7$ ;CHECK IF ZERO
5563 034410 005037 001124          CLR $GDDAT ;LOAD EXPECTED CONTENTS
5564 034414 012737 067553 001322    MOV #EM1022, EM3N+2 ;LOAD ERROR MESSAGE
5565 034422 104003          ERROR 3
5566 034424 016237 000014 001126 7$: MOV RKER(R2), $BDDAT ;STORE ERROR REG
5567 034432 001406          BEQ 8$ ;CHECK IF ZERO
5568 034434 005037 001124          CLR $GDDAT ;LOAD EXPECTED CONTENTS
5569 034440 012737 067611 001322    MOV #EM1023, EM3N+2 ;LOAD ERROR MESSAGE
5570 034446 104003          ERROR 3
5571 034450 016237 000020 001126 8$: MOV RKDCYL(R2), $BDDAT ;STORE CYLINDER ADD REG
5572 034456 001406          BEQ 10$ ;CHECK IF ZERO
5573 034460 005037 001124          CLR $GDDAT ;LOAD EXPECTED RESULTS
5574 034464 012737 067706 001322    MOV #EM1025, EM3N+2 ;LOAD ERROR MESSAGE
5575 034472 104003          ERROR 3
5576 034474 016237 000026 001126 10$: MOV RKMRI(R2), $BDDAT ;STORE MAINTENANCE REG 1
5577 034502 012737 002000 001124    MOV #MEWD, $GDDAT ;LOAD EXPECTED CONTENTS
5578 034510 032737 020000 001126    BIT #ECCW, $BDDAT
5579 034516 001403          BEQ 11$
5580 034520 052737 020000 001124    BIS #ECCW, $GDDAT
5581 034526 023737 001124 001126 11$: CMP $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
5582 034534 001404          BEQ 12$ ;YES, CONTINUE TEST
5583 034536 012737 067733 001322    MOV #EM1026, EM3N+2 ;LOAD ERROR MESSAGE
5584 034544 104003          ERROR 3
5585 034546 016237 000032 001126 12$: MOV RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
5586 034554 001406          BEQ 13$ ;CHECK IF ZERO
5587 034556 005037 001124          CLR $GDDAT ;LOAD EXPECTED CONTENTS
5588 034562 012737 070011 001322    MOV #EM1030, EM3N+2 ;LOAD ERROR MESSAGE
5589 034570 104003          ERROR 3
5590 034572 016237 000030 001126 13$: MOV RKECPS(R2), $BDDAT ;STORE ECC POSITION REG:
5591 034600 022737 004066 001126    CMP #4066, $BDDAT ;CHECK IF ECC POSITION CORRECT
5592 034606 001407          BEQ TST44 ;:YES, GO TO NEXT TEST
5593 034610 012737 004066 001124    MOV #4066, $GDDAT ;LOAD EXPECTED CONTENTS
5594 034616 012737 067764 001322    MOV #EM1029, EM3N+2 ;LOAD ERROR MESSAGE
5595 034624 104003          ERROR 3

```

```

5596
5597
5598 :*****
5599 :*TEST 44 REGISTER INTERACTION USING MAINT REG 2
5600 :*
5601 :* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
5602 :* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
5603 :* REGISTER 2 TO 177777 AND NO INTERACTION TAKES PLACE.
5604 :*****
5605 TST44: SCOPE
5606 MOV #500, $TIMES ;:DO 500. ITERATIONS
5607 MOV $BASE, R2 ;:LOAD RK611 BASE ADDRESS
5608 MOV #177777, CONFIG ;:LOAD CONFIGURATION WORD
5609 MOV #EM12, EM3N ;:LOAD ERROR MESSAGE

```



5610	034656	012762	100000	000000	MOV	#CCLR,RKCS1(R2)	;CLEAR RK611 WITH CONTROLLER CLEAR
5611	034664	005062	000002		CLR	RKWC(R2)	;CLEAR WORD COUNT REG.
5612	034670	012762	177777	000034	MOV	#177777,RKMR2(R2)	;WRITE RKMR2 WITH 177777
5613	034676	016237	000000	001126	MOV	RKCS1(R2),SBDDAT	;STORE COMMAND AND STATUS REG.1
5614	034704	022737	000200	001126	CMP	#RDY,\$BDDAT	;CHECK IF CS1 CORRECT
5615	034712	001407			BEQ	1\$	;YES,CONTINUE
5616	034714	012737	000200	001124	MOV	#RDY,\$GDDAT	;LOAD EXPECTED RESULTS
5617	034722	012737	067376	001322	MOV	#EM1017,EM3N+2	;LOAD ERROR MESSAGE
5618	034730	104003			ERROR	3	
5619	034732	016237	000004	001126	1\$: MOV	RKBA(R2),SBDDAT	;STORE BUS ADD REG.
5620	034740	001406			BEQ	2\$	;CHECK IF ZERO
5621	034742	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5622	034746	012737	067447	001322	MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
5623	034754	104003			ERROR	3	
5624	034756	016237	000002	001126	2\$: MOV	RKWC(R2),SBDDAT	;STORE WORK COUNT REG.
5625	034764	001406			BEQ	3\$	;CHECK IF ZERO
5626	034766	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5627	034772	012737	067417	001322	MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
5628	035000	104003			ERROR	3	
5629	035002	016237	000006	001126	3\$: MOV	RKDA(R2),SBDDAT	;STORE DISK ADD REG
5630	035010	001406			BEQ	4\$	;CHECK IF ZERO
5631	035012	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5632	035016	012737	067474	001322	MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
5633	035024	104003			ERROR	3	
5634	035026	016237	000016	001126	4\$: MOV	RKASOF(R2),SBDDAT	;STORE ATTENTION SUMMARY/OFFSET REG
5635	035034	001406			BEQ	5\$	;CHECK IF ZERO
5636	035036	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5637	035042	012737	067640	001322	MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
5638	035050	104003			ERROR	3	
5639	035052	016237	000010	001126	5\$: MOV	RKCS2(R2),SBDDAT	;STORE COMMAND AND STATUS REG. 2
5640	035060	022737	000100	001126	CMP	#IR,\$BDDAT	;CHECK IF CS2 CORRECT
5641	035066	001407			BEQ	6\$	;YES,CONTINUE
5642	035070	012737	000100	001124	MOV	#IR,\$GDDAT	;LOAD EXPECTED CONTENTS
5643	035076	012737	067532	001322	MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
5644	035104	104003			ERROR	3	
5645	035106	016237	000012	001126	6\$: MOV	RKDS(R2),SBDDAT	;STORE DRIVE STATUS REG.
5646	035114	001406			BEQ	7\$	;CHECK IF ZERO
5647	035116	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5648	035122	012737	067553	001322	MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
5649	035130	104003			ERROR	3	
5650	035132	016237	000014	001126	7\$: MOV	RKER(R2),SBDDAT	;STORE ERROR REG
5651	035140	001406			BEQ	8\$	;CHECK IF ZERO
5652	035142	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
5653	035146	012737	067611	001322	MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
5654	035154	104003			ERROR	3	
5655	035156	016237	000020	001126	8\$: MOV	RKDCYL(R2),SBDDAT	;STORE CYLINDER ADD REG
5656	035164	001406			BEQ	10\$	;CHECK IF ZERO
5657	035166	005037	001124		CLR	\$GDDAT	;LOAD EXPECTED RESULTS
5658	035172	012737	067706	001322	MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
5659	035200	104003			ERROR	3	
5660	035202	016237	000026	001126	10\$: MOV	RKMR1(R2),SBDDAT	;STORE MAINTENANCE REG 1
5661	035210	012737	002000	001124	MOV	#MEWD,\$GDDAT	;LOAD EXPECTED CONTENTS
5662	035216	032737	020000	001126	BIT	#ECCW,\$BDDAT	
5663	035224	001403			BEQ	11\$	
5664	035226	052737	020000	001124	BIS	#ECCW,\$GDDAT	
5665	035234	023737	001124	001126	11\$: CMP	\$GDDAT,\$BDDAT	;CHECK IF MR1 CORRECT



```
5666 035242 001404 BEQ 12$ ;YES,CONTINUE TEST
5667 035244 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
5668 035252 104003 ERROR 3
5669 035254 016237 000032 001126 12$: MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
5670 035262 001406 BEQ 13$ ;CHECK IF ZERO
5671 035264 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5672 035270 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5673 035276 104003 ERROR 3
5674 035300 016237 000030 001126 13$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG:
5675 035306 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION CORRECT
5676 035314 001407 BEQ TST45 ;:YES,GO TO NEXT TEST
5677 035316 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED CONTENTS
5678 035324 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5679 035332 104003 ERROR 3
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689 035334 000004 TST45: SCOPE
5690 035336 012737 000764 001200 MOV #500, $TIMES ;:DO 500. ITERATIONS
5691 035344 013702 001270 MOV $BASE,R2 ;LOAD RK611 BASE ADDRESS
5692 035350 012737 177777 002010 MOV #177777,CONFIG ;LOAD CONFIGURATION WORD
5693 035356 012737 064242 001320 MOV #EM13,EM3N ;LOAD ERROR MESSAGE
5694 035364 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
5695 035372 005062 000002 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
5696 035376 012762 177777 000036 MOV #177777,RKMR3(R2) ;WRITE RKMR3 WITH 177777
5697 035404 016237 000000 001126 MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
5698 035412 022737 000200 001126 CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
5699 035420 001407 BEQ 1$ ;YES,CONTINUE
5700 035422 012737 000200 001124 MOV #RDY,$GDDAT ;LOAD EXPECTED RESULTS
5701 035430 012737 067376 001322 MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
5702 035436 104003 ERROR 3
5703 035440 016237 000004 001126 1$: MOV RKBA(R2),$BDDAT ;STORE BUS ADD REG.
5704 035446 001406 BEQ 2$ ;CHECK IF ZERO
5705 035450 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5706 035454 012737 067447 001322 MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
5707 035462 104003 ERROR 3
5708 035464 016237 000002 001126 2$: MOV RKWC(R2),$BDDAT ;STORE WORK COUNT REG.
5709 035472 001406 BEQ 3$ ;CHECK IF ZERO
5710 035474 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5711 035500 012737 067417 001322 MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
5712 035506 104003 ERROR 3
5713 035510 016237 000006 001126 3$: MOV RKDA(R2),$BDDAT ;STORE DISK ADD REG
5714 035516 001406 BEQ 4$ ;CHECK IF ZERO
5715 035520 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5716 035524 012737 067474 001322 MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
5717 035532 104003 ERROR 3
5718 035534 016237 000016 001126 4$: MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG
5719 035542 001406 BEQ 5$ ;CHECK IF ZERO
5720 035544 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
5721 035550 012737 067640 001322 MOV #EM1024,EM3N+2 ;LOAD ERROR MESSAGE
```



5722	035556	104003				ERROR	3	
5723	035560	016237	000010	001126	5\$:	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG. 2
5724	035566	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
5725	035574	001407				BEQ	6\$	:YES, CONTINUE
5726	035576	012737	000100	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
5727	035604	012737	067532	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
5728	035612	104003				ERROR	3	
5729	035614	016237	000012	001126	6\$:	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG.
5730	035622	001406				BEQ	7\$	:CHECK IF ZERO
5731	035624	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5732	035630	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
5733	035636	104003				ERROR	3	
5734	035640	016237	000014	001126	7\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
5735	035646	001406				BEQ	8\$	:CHECK IF ZERO
5736	035650	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5737	035654	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
5738	035662	104003				ERROR	3	
5739	035664	016237	000020	001126	8\$:	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
5740	035672	001406				BEQ	10\$	:CHECK IF ZERO
5741	035674	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED RESULTS
5742	035700	012737	067706	001322		MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
5743	035706	104003				ERROR	3	
5744	035710	016237	000026	001126	10\$:	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG 1
5745	035716	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED CONTENTS
5746	035724	032737	020000	001126		BIT	#ECCW, \$BDDAT	
5747	035732	001403				BEQ	11\$	
5748	035734	052737	020000	001124		BIS	#ECCW, \$GDDAT	
5749	035742	023737	001124	001126	11\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
5750	035750	001404				BEQ	12\$	:YES, CONTINUE TEST
5751	035752	012737	067733	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
5752	035760	104003				ERROR	3	
5753	035762	016237	000032	001126	12\$:	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
5754	035770	001406				BEQ	13\$	:CHECK IF ZERO
5755	035772	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5756	035776	012737	070011	001322		MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
5757	036004	104003				ERROR	3	
5758	036006	016237	000030	001126	13\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG:
5759	036014	022737	004066	001126		CMP	#4066, \$BDDAT	:CHECK IF ECC POSITION CORRECT
5760	036022	001407				BEQ	TST46	:::YES, GO TO NEXT TEST
5761	036024	012737	004066	001124		MOV	#4066, \$GDDAT	:LOAD EXPECTED CONTENTS
5762	036032	012737	067764	001322		MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
5763	036040	104003				ERROR	3	

5764  
5765  
5766  
5767  
5768  
5769  
5770  
5771  
5772  
5773  
5774  
5775  
5776  
5777

```
*****  
*TEST 46 REGISTER INTERACTION WITH DISK CYLINDER (PART 1)  
*  
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
* CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS  
* ARE CORRECT AND NO INTERACTION TAKES PLACE.  
*  
* 000000 000010 000200 004000 100000  
* 000001 000020 000400 010000  
* 000002 000040 001000 020000  
* 000004 000100 002000 040000  
*  
*****
```



```
5778  
5779 036042 000004  
5780 036044 012737 000764 001200  
5781 036052 012701 000021  
5782 036056 012737 000001 002010  
5783 036064 012737 064540 001320  
5784 036072 012762 100000 000000  
5785 036100 012737 036106 001110  
5786  
5787  
5788 036106  
5789 036106 005062 000002  
5790 036112 013762 002010 000020  
5791 036120 016237 000020 001126  
5792 036126 013737 002010 001124  
5793 036134 042737 176000 001124  
5794 036142 023737 001124 001126  
5795 036150 001404  
5796 036152 012737 067354 001322  
5797 036160 104003  
5798 036162  
5799 036162 016237 000000 001126  
5800 036170 022737 000200 001126  
5801 036176 001407  
5802 036200 012737 000100 001124  
5803 036206 012737 067376 001322  
5804 036214 104003  
5805 036216  
5806 036216 016237 000004 001126  
5807 036224 001406  
5808 036226 005037 001124  
5809 036232 012737 067447 001322  
5810 036240 104003  
5811 036242  
5812 036242 016237 000002 001126  
5813 036250 001406  
5814 036252 005037 001124  
5815 036256 012737 067417 001322  
5816 036264 104003  
5817 036266  
5818 036266 016237 000006 001126  
5819 036274 001406  
5820 036276 005037 001124  
5821 036302 012737 067474 001322  
5822 036310 104003  
5823 036312  
5824 036312 016237 000016 001126  
5825 036320 001406  
5826 036322 005037 001124  
5827 036326 012737 067640 001322  
5828 036334 104003  
5829 036336  
5830 036336 016237 000010 001126  
5831 036344 022737 000100 001126  
5832 036352 001407  
5833 036354 012737 000100 001124
```

\*\*\*\*\*  
TST46: SCOPE  
MOV #500, \$TIMES ;DO 500. ITERATIONS  
MOV #17, R1 ;LOAD NUMBER OF PATTERNS  
MOV #000001, CONFIG ;LOAD INITIAL CONFIGURATION  
MOV #EM19, EM3N ;LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$:  
CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MOV CONFIG, RKDCYL(R2) ;WRITE RKDCYL  
MOV RKDCYL(R2), \$BDDAT ;STORE RKDCYL  
MOV CONFIG, \$GDDAT ;PREPARE EXPECTED RESULTS  
BIC #176000, \$GDDAT ;INITIALIZE READ ONLY BITS  
CMP \$GDDAT, \$BDDAT ;CHECK IF RKDCYL CORRECT  
BEQ 2\$ ;YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1016, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

2\$:  
MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG. 1  
CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT  
BEQ 3\$ ;YES, CONTINUE  
MOV #IR, \$GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

3\$:  
MOV RKBA(R2), \$BDDAT ;STORE BUS AND REG  
BEQ 4\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

4\$:  
MOV RKWC(R2), \$BDDAT ;STORE WORD COUNT REG  
BEQ 5\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

5\$:  
MOV RKDA(R2), \$BDDAT ;STORE DISK AVERAGE REG  
BEQ 6\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1020, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

6\$:  
MOV RKASOF(R2), \$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.  
BEQ 7\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1024, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

7\$:  
MOV RKCS2(R2), \$BDDAT ;STORE COMMAND AND STATUS REG.2  
CMP #IR, \$BDDAT ;CHECK IF CS2 CORRECT  
BEQ 8\$ ;YES, CONTINUE  
MOV #IR, \$GDDAT ;LOAD EXPECTED CONTENTS



```

5834 036362 012737 067532 001322      MOV      #EM1021,EM3N+2 ;LOAD ERROR MESSAGE
5835 036370 104003      ERROR    3
5836 036372      8$:
5837 036372 016237 000012 001126      MOV      RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG
5838 036400 001406      BEQ      9$ ;CHECK IF ZERO
5839 036402 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5840 036406 012737 067553 001322      MOV      #EM1022,EM3N+2 ;LOAD ERROR MESSAGE
5841 036414 104003      ERROR    3
5842 036416 016237 000014 001126      9$:      MOV      RKER(R2), $BDDAT ;STORE ERROR REG
5843 036424 001406      BEQ      10$ ;CHECK IF ZERO
5844 036426 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5845 036432 012737 067611 001322      MOV      #EM1023,EM3N+2 ;LOAD ERROR MESSAGE
5846 036440 104003      ERROR    3
5847 036442      10$:
5848 036442 016237 000026 001126      MOV      RKMR1(R2), $BDDAT ;STORE MAINTENANCE REG.1
5849 036450 012737 002000 001124      MOV      #MEWD, $GDDAT ;LOAD EXPECTED MR1
5850 036456 032737 020000 001126      BIT      #ECCW, $BDDAT
5851 036464 001403      BEQ      13$
5852 036466 052737 020000 001124      BIS      #ECCW, $GDDAT
5853 036474 023737 001124 001126      13$:     CMP      $GDDAT, $BDDAT ;CHECK IF MR1 CORRECT
5854 036502 001404      BEQ      14$ ;YES, ISSUE CONTROLLER CLEAR
5855 036504 012737 067733 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
5856 036512 104003      ERROR    3
5857 036514      14$:
5858 036514 016237 000032 001126      MOV      RKECPT(R2), $BDDAT ;STORE ECC PATTERN REG.
5859 036522 001406      BEQ      15$ ;CHECK IF ZERO
5860 036524 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5861 036530 012737 070011 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
5862 036536 104003      ERROR    3
5863 036540 016237 000030 001126      15$:     MOV      RKECPS(R2), $BDDAT ;STORE ECC POSITION REG.
5864 036546 012737 074066 001124      16$:     MOV      #4066, $GDDAT ;USE 4066
5865 036554 023737 001124 001126      17$:     CMP      $GDDAT, $BDDAT ;CHECK IF ECC POSITION CORRECT
5866 036562 001404      BEQ      18$ ;YES, INITIALIZE RK611
5867 036564 012737 067764 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
5868 036572 104003      ERROR    3
5869 036574 016237 000020 002014      18$:     MOV      RKDCYL(R2), PREREG ;GET PREVIOUS CONTENTS
5870 036602 012762 100000 000000      MOV      #CCLR, RKCS1(R2) ;CLEAR RK611 CONTROLLER
5871 036610 016237 000020 001126      MOV      RKDCYL(R2), $BDDAT ;GET CURRENT VALUE
5872 036616 005037 001124      CLR      $GDDAT ;LOAD EXPECTED CONTENTS
5873 036622 023737 001124 001126      CMP      $GDDAT, $BDDAT ;CHECK IF RKDCYL CORRECT
5874 036630 001407      BEQ      19$ ;YES, CHECK IF FINISHED
5875 036632 012737 063465 001330      MOV      #EM3, EM4N ;LOAD ERROR MESSAGE
5876 036640 012737 067354 001332      MOV      #EM1016,EM4N+2
5877 036646 104004      ERROR    4
5878 036650 104415      19$:     SCOP1   ;CHECK IF LOOP ON ERROR
5879 036652 000241      CLC      ;SHIFT IN ZERO
5880 036654 006137 002010      ROL      CONFIG
5881 036660 005301      DEC      R1 ;CHECK IF FINISHED
5882 036662 001402      BEQ      TST47 ;:YES, GO ON TO NEXT TEST
5883 036664 000137 036106      JMP      1$
5884
5885
5886
5887
5888
5889

```

```

*****
*TEST 47 REGISTER INTERACTION WITH DISK CYLINDER (PART 2)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK

```



```

5890
5891
5892
5893
5894
5895
5896
5897
5898
5899 036670 000004
5900 036672 012737 000764 001200
5901 036700 012701 000021
5902 036704 012737 177776 002010
5903 036712 012737 064540 001320
5904 036720 012762 100000 000000
5905 036726 012737 036734 001110
5906
5907
5908 036734
5909 036734 005062 000002
5910 036740 013762 002010 000020
5911 036746 016237 000020 001126
5912 036754 013737 002010 001124
5913 036762 042737 176000 001124
5914 036770 023737 001124 001126
5915 036776 001404
5916 037000 012737 067354 001322
5917 037006 104003
5918 037010
5919 037010 016237 000000 001126
5920 037016 022737 000200 001126
5921 037024 001407
5922 037026 012737 000100 001124
5923 037034 012737 067376 001322
5924 037042 104003
5925 037044
5926 037044 016237 000004 001126
5927 037052 001406
5928 037054 005037 001124
5929 037060 012737 067447 001322
5930 037066 104003
5931 037070
5932 037070 016237 000002 001126
5933 037076 001406
5934 037100 005037 001124
5935 037104 012737 067417 001322
5936 037112 104003
5937 037114
5938 037114 016237 000006 001126
5939 037122 001406
5940 037124 005037 001124
5941 037130 012737 067474 001322
5942 037136 104003
5943 037140
5944 037140 016237 000016 001126
5945 037146 001406

```

```

:* CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
:* ARE CORRECT AND NO INTERACTION TAKES PLACE.

```

```

:* 177777 177767 177577 173777 077777
:* 177776 177757 177377 167777
:* 177775 177737 176777 157777
:* 177773 177677 175777 137777

```

```

*****
TST47: SCOPE
MOV #500,$TIMES ;;DO 500. ITERATIONS
MOV #17,R1 ;LOAD NUMBER OF PATTERNS
MOV #177776,CONFIG ;LOAD INITIAL CONFIGURATION
MOV #EM19,EM3N ;LOAD ERROR MESSAGE
MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #1$,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
; SUBTEST LOOP

```

```

1$: CLR RKWC(R2) ;CLEAR WORD COUNT REG.
MOV CONFIG,RKDCYL(R2) ;WRITE RKDCYL
MOV RKDCYL(R2),$BDDAT ;STORE RKDCYL
MOV CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
BIC #176000,$GDDAT ;INITIALIZE READ ONLY BITS
CMP $GDDAT,$BDDAT ;CHECK IF RKDCYL CORRECT
BEQ 2$ ;YES,TEST IF ANY OTHER REG MODIFIED
MOV #EM1016,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

2$: MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG. 1
CMP #RDY,$BDDAT ;CHECK IF CS1 CORRECT
BEQ 3$ ;YES, CONTINUE
MOV #IR,$GDDAT ;LOAD EXPECTED RESULTS
MOV #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

3$: MOV RKBA(R2),$BDDAT ;STORE BUS AND REG
BEQ 4$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1019,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

4$: MOV RKWC(R2),$BDDAT ;STORE WORD COUNT REG
BEQ 5$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1018,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

5$: MOV RKDA(R2),$BDDAT ;STORE DISK AVERAGE REG
BEQ 6$ ;CHECK IF ZERO
CLR $GDDAT ;LOAD EXPECTED CONTENTS
MOV #EM1020,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

```

```

6$: MOV RKASOF(R2),$BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG.
BEQ 7$ ;CHECK IF ZERO

```



5946	037150	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5947	037154	012737	067640	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
5948	037162	104003			ERROR	3	
5949	037164				7\$:		
5950	037164	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
5951	037172	022737	000100	001126	CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
5952	037200	001407			BEQ	8\$	:YES, CONTINUE
5953	037202	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
5954	037210	012737	067532	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
5955	037216	104003			ERROR	3	
5956	037220				8\$:		
5957	037220	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
5958	037226	001406			BEQ	9\$	:CHECK IF ZERO
5959	037230	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5960	037234	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
5961	037242	104003			ERROR	3	
5962	037244	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT :STORE ERROR REG
5963	037252	001406			BEQ	10\$	:CHECK IF ZERO
5964	037254	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5965	037260	012737	067611	001322	MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
5966	037266	104003			ERROR	3	
5967	037270				10\$:		
5968	037270	016237	000026	001126	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
5969	037276	012737	002000	001124	MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
5970	037304	032737	020000	001126	BIT	#ECCW, \$BDDAT	
5971	037312	001403			BEQ	13\$	
5972	037314	052737	020000	001124	BIS	#ECCW, \$GDDAT	
5973	037322	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT :CHECK IF MR1 CORRECT
5974	037330	001404			BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
5975	037332	012737	067733	001322	MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
5976	037340	104003			ERROR	3	
5977	037342				14\$:		
5978	037342	016237	000032	001126	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
5979	037350	001406			BEQ	15\$	:CHECK IF ZERO
5980	037352	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5981	037356	012737	070011	001322	MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
5982	037364	104003			ERROR	3	
5983	037366	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT :STORE ECC POSITION REG.
5984	037374	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT :USE 4066
5985	037402	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT :CHECK IF ECC POSITION CORRECT
5986	037410	001404			BEQ	18\$	:YES, INITIALIZE RK611
5987	037412	012737	067764	001322	MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
5988	037420	104003			ERROR	3	
5989	037422	016237	000020	002014	18\$:	MOV	RKDCYL(R2), PREREG :GET PREVIOUS CONTENTS
5990	037430	012762	100000	000000	MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
5991	037436	016237	000020	001126	MOV	RKDCYL(R2), \$BDDAT	:GET CURRENT VALUE
5992	037444	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
5993	037450	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	:CHECK IF RKDCYL CORRECT
5994	037456	001407			BEQ	19\$	:YES, CHECK IF FINISHED
5995	037460	012737	063465	001330	MOV	#EM3, EM4N	:LOAD ERROR MESSAGE
5996	037466	012737	067354	001332	MOV	#EM1016,EM4N+2	
5997	037474	104004			ERROR	4	
5998	037476	104415			19\$:	SCOP1	:CHECK IF LOOP ON ERROR
5999	037500	000261			SEC		:SHIFT IN ONE
6000	037502	006137	002010		ROL	CONFIG	
6001	037506	005301			DEC	R1	:CHECK IF FINISHED



6002 037510 001402  
6003 037512 000137 036734  
6004  
6005  
6006  
6007  
6008  
6009  
6010  
6011  
6012  
6013  
6014  
6015  
6016  
6017  
6018  
6019 037516 000004  
6020 037520 012737 000764 001200  
6021 037526 012701 000021  
6022 037532 005037 002010  
6023 037536 012737 064540 001320  
6024 037544 012762 100000 000000  
6025 037552 012737 037560 001110  
6026  
6027  
6028 037560  
6029 037560 005062 000002  
6030 037564 013762 002010 000020  
6031 037572 016237 000020 001126  
6032 037600 013737 002010 001124  
6033 037606 042737 176000 001124  
6034 037614 023737 001124 001126  
6035 037622 001404  
6036 037624 012737 067354 001322  
6037 037632 104003  
6038 037634  
6039 037634 016237 000000 001126  
6040 037642 022737 000200 001126  
6041 037650 001407  
6042 037652 012737 000100 001124  
6043 037660 012737 067376 001322  
6044 037666 104003  
6045 037670  
6046 037670 016237 000004 001126  
6047 037676 001406  
6048 037700 005037 001124  
6049 037704 012737 067447 001322  
6050 037712 104003  
6051 037714  
6052 037714 016237 000002 001126  
6053 037722 001406  
6054 037724 005037 001124  
6055 037730 012737 067417 001322  
6056 037736 104003  
6057 037740

BEQ TST50 ;:YES, GO ON TO NEXT TEST  
JMP 1\$

```
*****  
*TEST 50 REGISTER INTERACTION WITH DISK CYLINDER (PART 3)  
*  
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
* CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS  
* ARE CORRECT AND NO INTERACTION TAKES PLACE.  
*  
* 000001 000037 000777 017777 000000  
* 000003 000077 001777 037777  
* 000007 000177 003777 077777  
* 000017 000377 007777 177777  
*****
```

```
TST50: SCOPE  
MOV #500, $TIMES ;:DO 500. ITERATIONS  
MOV #17, R1 ;:LOAD NUMBER OF PATTERNS  
CLR CONFIG ;:LOAD INITIAL CONFIGURATION  
MOV #EM19, EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1$, $LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP  
  
1$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG, RKDCYL(R2) ;:WRITE RKDCYL  
MOV RKDCYL(R2), $BDDAT ;:STORE RKDCYL  
MOV CONFIG, $GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #176000, $GDDAT ;:INITIALIZE READ ONLY BITS  
CMP $GDDAT, $BDDAT ;:CHECK IF RKDCYL CORRECT  
BEQ 2$ ;:YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1016, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
2$:  
MOV RKCS1(R2), $BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY, $BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3$ ;:YES, CONTINUE  
MOV #IR, $GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
3$:  
MOV RKBA(R2), $BDDAT ;:STORE BUS AND REG  
BEQ 4$ ;:CHECK IF ZERO  
CLR $GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
4$:  
MOV RKWC(R2), $BDDAT ;:STORE WORD COUNT REG  
BEQ 5$ ;:CHECK IF ZERO  
CLR $GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
5$:
```



6058	037740	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
6059	037746	001406				BEQ	6\$	:CHECK IF ZERO
6060	037750	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6061	037754	012737	067474	001322		MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
6062	037762	104003				ERROR	3	
6063	037764				6\$:			
6064	037764	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
6065	037772	001406				BEQ	7\$	:CHECK IF ZERO
6066	037774	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6067	040000	012737	067640	001322		MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
6068	040006	104003				ERROR	3	
6069	040010				7\$:			
6070	040010	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
6071	040016	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
6072	040024	001407				BEQ	8\$	:YES, CONTINUE
6073	040026	012737	000100	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
6074	040034	012737	067532	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
6075	040042	104003				ERROR	3	
6076	040044				8\$:			
6077	040044	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
6078	040052	001406				BEQ	9\$	:CHECK IF ZERO
6079	040054	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6080	040060	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
6081	040066	104003				ERROR	3	
6082	040070	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
6083	040076	001406				BEQ	10\$	:CHECK IF ZERO
6084	040100	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6085	040104	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
6086	040112	104003				ERROR	3	
6087	040114				10\$:			
6088	040114	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
6089	040122	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
6090	040130	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6091	040136	001403				BEQ	13\$	
6092	040140	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6093	040146	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
6094	040154	001404				BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
6095	040156	012737	067733	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
6096	040164	104003				ERROR	3	
6097	040166				14\$:			
6098	040166	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
6099	040174	001406				BEQ	15\$	:CHECK IF ZERO
6100	040176	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6101	040202	012737	070011	001322		MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
6102	040210	104003				ERROR	3	
6103	040212	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
6104	040220	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
6105	040226	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
6106	040234	001404				BEQ	18\$	:YES, INITIALIZE RK611
6107	040236	012737	067764	001322		MOV	#EM1029, EM3N+2	:LOAD ERROR MESSAGE
6108	040244	104003				ERROR	3	
6109	040246	016237	000020	002014	18\$:	MOV	RKDCYL(R2), PREREG	:GET PREVIOUS CONTENTS
6110	040254	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
6111	040262	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	:GET CURRENT VALUE
6112	040270	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6113	040274	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	:CHECK IF RKDCYL CORRECT



6114	040302	001407			BEQ	19\$		:YES, CHECK IF FINISHED
6115	040304	012737	063465	001330	MOV	#EM3,EM4N		:LOAD ERROR MESSAGE
6116	040312	012737	067354	001332	MOV	#EM1016,EM4N+2		
6117	040320	104004			ERROR	4		
6118	040322	104415			19\$: SCOPI			:CHECK IF LOOP ON ERROR
6119	040324	000261			SEC			:SHIFT IN ONE
6120	040326	006137	002010		ROL	CONFIG		
6121	040332	005301			DEC	R1		:CHECK IF FINISHED
6122	040334	001402			BEQ	TST51		:;YES, GO ON TO NEXT TEST
6123	040336	000137	037560		JMP	1\$		

\*\*\*\*\*  
 \*TEST 51 REGISTER INTERACTION WITH DISK CYLINDER (PART 4)  
 \*\*\*\*\*

ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
 WRITE THE WORD COUNT REGISTER TO ZERO. WRITE THE DISK  
 CYLINDER ADDRESS WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS  
 ARE CORRECT AND NO INTERACTION TAKES PLACE.

*	100000	174000	177600	177770	000000
*	140000	176000	177700	177774	
*	160000	177000	177740	177776	
*	170000	177400	177760	177777	

\*\*\*\*\*  
 TST51: SCOPE  
 MOV #500,\$TIMES ;;DO 500. ITERATIONS  
 MOV #17,R1 ;LOAD NUMBER OF PATTERNS  
 CLR CONFIG ;LOAD INITIAL CONFIGURATION  
 MOV #EM19,EM3N ;LOAD ERROR MESSAGE  
 MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
 MOV #1\$,\$LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
 ; SUBTEST LOOP

6139	040342	000004			1\$: CLR	RKWC(R2)		:CLEAR WORD COUNT REG.
6140	040344	012737	000764	001200	MOV	CONFIG,RKDCYL(R2)		:WRITE RKDCYL
6141	040352	012701	000021		MOV	RKDCYL(R2),\$BDDAT		:STORE RKDCYL
6142	040356	005037	002010		MOV	CONFIG,\$GDDAT		:PREPARE EXPECTED RESULTS
6143	040362	012737	064540	001320	BIC	#176000,\$GDDAT		:INITIALIZE READ ONLY BITS
6144	040370	012762	100000	000000	CMP	\$GDDAT,\$BDDAT		:CHECK IF RKDCYL CORRECT
6145	040376	012737	040404	001110	BEQ	2\$		:YES,TEST IF ANY OTHER REG MODIFIED
6146					MOV	#EM1016,EM3N+2		:LOAD ERROR MESSAGE
6147					ERROR	3		
6148	040404				2\$: MOV	RKCS1(R2),\$BDDAT		:STORE COMMAND AND STATUS REG. 1
6149	040404	005062	000002		CMP	#RDY,\$BDDAT		:CHECK IF CS1 CORRECT
6150	040410	013762	002010	000020	BEQ	3\$		:YES, CONTINUE
6151	040416	016237	000020	001126	MOV	#IR,\$GDDAT		:LOAD EXPECTED RESULTS
6152	040424	013737	002010	001124	MOV	#EM1017,EM3N+2		:LOAD ERROR MESSAGE
6153	040432	042737	176000	001124	ERROR	3		
6154	040440	023737	001124	001126	3\$: MOV	RKBA(R2),\$BDDAT		:STORE BUS AND REG
6155	040446	001404			BEQ	4\$		:CHECK IF ZERO
6156	040450	012737	067354	001322	CLR	\$GDDAT		:LOAD EXPECTED CONTENTS
6157	040456	104003			MOV	#EM1019,EM3N+2		:LOAD ERROR MESSAGE
6158	040460							
6159	040460	016237	000000	001126				
6160	040466	022737	000200	001126				
6161	040474	001407						
6162	040476	012737	000100	001124				
6163	040504	012737	067376	001322				
6164	040512	104003						
6165	040514							
6166	040514	016237	000004	001126				
6167	040522	001406						
6168	040524	005037	001124					
6169	040530	012737	067447	001322				



6170	040536	104003				ERROR	3	
6171	040540				4\$:			
6172	040540	016237	000002	001126		MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG
6173	040546	001406				BEQ	5\$	:CHECK IF ZERO
6174	040550	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6175	040554	012737	067417	001322		MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
6176	040562	104003				ERROR	3	
6177	040564				5\$:			
6178	040564	016237	000006	001126		MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
6179	040572	001406				BEQ	6\$	:CHECK IF ZERO
6180	040574	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6181	040600	012737	067474	001322		MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
6182	040606	104003				ERROR	3	
6183	040610				6\$:			
6184	040610	016237	000016	001126		MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
6185	040616	001406				BEQ	7\$	:CHECK IF ZERO
6186	040620	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6187	040624	012737	067640	001322		MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
6188	040632	104003				ERROR	3	
6189	040634				7\$:			
6190	040634	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
6191	040642	022737	000100	001126		CMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
6192	040650	001407				BEQ	8\$	:YES, CONTINUE
6193	040652	012737	000100	001124		MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
6194	040660	012737	067532	001322		MOV	#EM1021, EM3N+2	:LOAD ERROR MESSAGE
6195	040666	104003				ERROR	3	
6196	040670				8\$:			
6197	040670	016237	000012	001126		MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
6198	040676	001406				BEQ	9\$	:CHECK IF ZERO
6199	040700	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6200	040704	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
6201	040712	104003				ERROR	3	
6202	040714	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
6203	040722	001406				BEQ	10\$	:CHECK IF ZERO
6204	040724	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6205	040730	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
6206	040736	104003				ERROR	3	
6207	040740				10\$:			
6208	040740	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG.1
6209	040746	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
6210	040754	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6211	040762	001403				BEQ	13\$	
6212	040764	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6213	040772	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
6214	041000	001404				BEQ	14\$	:YES, ISSUE CONTROLLER CLEAR
6215	041002	012737	067733	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
6216	041010	104003				ERROR	3	
6217	041012				14\$:			
6218	041012	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
6219	041020	001406				BEQ	15\$	:CHECK IF ZERO
6220	041022	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6221	041026	012737	070011	001322		MOV	#EM1030, EM3N+2	:LOAD ERROR MESSAGE
6222	041034	104003				ERROR	3	
6223	041036	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
6224	041044	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
6225	041052	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT



```

6226 041060 001404          BEQ      18$          ;YES,INITIALIZE RK611
6227 041062 012737 067764 001322  MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
6228 041070 104003          ERROR    3
6229 041072 016237 000020 002014 18$:  MOV      RKDCYL(R2),PREREG ;GET PREVIOUS CONTENTS
6230 041100 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
6231 041106 016237 000020 001126  MOV      RKDCYL(R2),$BDDAT ;GET CURRENT VALUE
6232 041114 005037 001124          CLR      $GDDAT ;LOAD EXPECTED CONTENTS
6233 041120 023737 001124 001126  CMP      $GDDAT,$BDDAT ;CHECK IF RKDCYL CORRECT
6234 041126 001407          BEQ      19$          ;YES, CHECK IF FINISHED
6235 041130 012737 063465 001330  MOV      #EM3,EM4N ;LOAD ERROR MESSAGE
6236 041136 012737 067354 001332  MOV      #EM1016,EM4N+2
6237 041144 104004          ERROR    4
6238 041146 104415          19$:  SCOP1 ;CHECK IF LOOP ON ERROR
6239 041150 000261          SEC ;SHIFT IN ONE
6240 041152 006037 002010  ROR      CONFIG
6241 041156 005301          DEC      R1 ;CHECK IF FINISHED
6242 041160 001402          BEQ      TST52 ;:YES, GO ON TO NEXT TEST
6243 041162 000137 040404  JMP      1$
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259 041166 000004          TST52: SCOPE
6260 041170 012737 000764 001200  MOV      #500,$TIMES ;:DO 500. ITERATIONS
6261 041176 012701 000021          MOV      #17,R1 ;LOAD NUMBER OF PATTERNS
6262 041202 012737 000001 002010  MOV      #000001,CONFIG ;LOAD INITIAL CONFIGURATION
6263 041210 012737 064601 001320  MOV      #EM20,EM3N ;LOAD ERROR MESSAGE
6264 041216 012762 100000 000000  MOV      #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
6265 041224 012737 041232 001110  MOV      #1,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
6266                                     ; SUBTEST LOOP
6267
6268 041232          1$:  CLR      RKWC(R2) ;CLEAR WORD COUNT REG.
6269 041232 005062 000002          MOV      CONFIG,RKMR1(R2) ;WRITE RKMR1
6270 041236 013762 002010 000026  MOV      RKMR1(R2),$BDDAT ;STORE RKMR1
6271 041244 016237 000026 001126  MOV      CONFIG,$GDDAT ;PREPARE EXPECTED RESULTS
6272 041252 013737 002010 001124  BIC      #PCA!PCD!ECCW!WRTGAT!RDGATE,$GDDAT ;INITIALIZE READ ONLY BITS
6273 041260 042737 174000 001124  BIS      #MEWD,$GDDAT
6274 041266 052737 002000 001124  BIT      #ECCW,$BDDAT
6275 041274 032737 020000 001126  BEQ      30$
6276 041302 001403          BIS      #ECCW,$GDDAT
6277 041304 052737 020000 001124  30$:  CMP      $GDDAT,$BDDAT ;CHECK IF RKMR1 CORRECT
6278 041312          BEQ      2$ ;YES,TEST IF ANY OTHER REG MODIFIED
6279 041312 023737 001124 001126  MOV      #EM1009,EM3N+2 ;LOAD ERROR MESSAGE
6280 041320 001404          2$
6281 041322 012737 067154 001322

```

```

*****
*TEST 52 REGISTER INTERACTION USING MAINT REG 1 (PART 1)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
* REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
* ARE CORRECT AND NO INTERACTION TAKES PLACE.
*
* 000000 000010 000200 004000 100000
* 000001 000020 000400 010000
* 000002 000040 001000 020000
* 000004 000100 002000 040000
*****

```







6338	041636				12\$:		
6339	041636	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT ; STORE ECC PATTERN REG.
6340	041644	001400				BEQ	15\$ ; CHECK IF ZERO
6341	041646	005037	001124			CLR	\$GDDAT ; LOAD EXPECTED CONTENTS
6342	041652	012737	070011	001322		MOV	#EM1030, EM3N+2 ; LOAD ERROR MESSAGE
6343	041660	104003				ERROR	3
6344	041662	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT ; STORE ECC POSITION REG.
6345	041670	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT ; USE 4066
6346	041676	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT ; CHECK IF ECC POSITION CORRECT
6347	041704	001404				BEQ	18\$ ; YES, INITIALIZE RK611
6348	041706	012737	067764	001322		MOV	#EM1029, EM3N+2 ; LOAD ERROR MESSAGE
6349	041714	104003				ERROR	3
6350	041716	016237	000026	002014	18\$:	MOV	RKMR1(R2), PREREG ; GET PREVIOUS CONTENTS
6351	041724	012762	100000	000000		MOV	#CCLR, RKCS1(R2) ; CLEAR RK611 CONTROLLER
6352	041732	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT ; GET CURRENT VALUE
6353	041740	012737	002000	001124		MOV	#MEWD, \$GDDAT ; LOAD EXPECTED CONTENTS
6354	041746	032737	020000	001126		BIT	#ECCW, \$BDDAT
6355	041754	001403				BEQ	35\$
6356	041756	052737	020000	001124		BIS	#ECCW, \$GDDAT
6357	041764				35\$:		
6358	041764	023737	001124	001126		CMP	\$GDDAT, \$BDDAT ; CHECK IF RKMR1 CORRECT
6359	041772	001407				BEQ	19\$ ; YES, CHECK IF FINISHED
6360	041774	012737	063465	001330		MOV	#EM3, EM4N ; LOAD ERROR MESSAGE
6361	042002	012737	067154	001332		MOV	#EM1009, EM4N+2
6362	042010	104004				ERROR	4
6363	042012	104415			19\$:	SCOPE	; CHECK IF LOOP ON ERROR
6364	042014	000241				CLC	; SHIFT IN ZERO
6365	042016	006137	002010			ROL	CONFIG
6366	042022	005301				DEC	R1 ; CHECK IF FINISHED
6367	042024	001402				BEQ	TST53 ; YES, GO ON TO NEXT TEST
6368	042026	000137	041232			JMP	1\$

```

*****
*TEST 53 REGISTER INTERACTION USING MAINT REG 1 (PART 2)
*
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
* WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
* REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
* ARE CORRECT AND NO INTERACTION TAKES PLACE.
*
* 177777 177767 177577 173777 077777
* 177776 177757 177377 167777
* 177775 177737 176777 157777
* 177773 177677 175777 137777
*****
    
```

6384	042032	000004			TST53:	SCOPE	
6385	042034	012737	000764	001200		MOV	#500, \$TIMES ; DO 500. ITERATIONS
6386	042042	012701	000021			MOV	#17, R1 ; LOAD NUMBER OF PATTERNS
6387	042046	012737	177776	002010		MOV	#177776, CONFIG ; LOAD INITIAL CONFIGURATION
6388	042054	012737	064601	001320		MOV	#EM20, EM3N ; LOAD ERROR MESSAGE
6389	042062	012762	100000	000000		MOV	#CCLR, RKCS1(R2) ; CLEAR RK611 WITH CONTROLLER CLEAR
6390	042070	012737	042076	001110		MOV	#1\$, \$LPERR ; LOAD LOOP ON ERROR LOCATION FOR
6391							; SUBTEST LOOP
6392							
6393	042076				1\$:		







6450	042422	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
6451	042430	104003				ERROR	3	
6452	042432	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
6453	042440	001406				BEQ	10\$	:CHECK IF ZERO
6454	042442	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6455	042446	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
6456	042454	104003				ERROR	3	
6457	042456				10\$:			
6458	042456	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
6459	042464	001406				BEQ	12\$	:CHECK IF ZERO
6460	042466	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6461	042472	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
6462	042500	104003				ERROR	3	
6463	042502				12\$:			
6464	042502	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
6465	042510	001406				BEQ	15\$	:CHECK IF ZERO
6466	042512	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6467	042516	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
6468	042524	104003				ERROR	3	
6469	042526	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
6470	042534	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	:USE 4066
6471	042542	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
6472	042550	001404				BEQ	18\$	:YES, INITIALIZE RK611
6473	042552	012737	067764	001322		MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
6474	042560	104003				ERROR	3	
6475	042562	016237	000026	002014	18\$:	MOV	RKMR1(R2), PREREG	:GET PREVIOUS CONTENTS
6476	042570	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
6477	042576	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	:GET CURRENT VALUE
6478	042604	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED CONTENTS
6479	042612	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6480	042620	001403				BEQ	35\$	
6481	042622	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6482	042630				35\$:			
6483	042630	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	:CHECK IF RKMR1 CORRECT
6484	042636	001407				BEQ	19\$	:YES, CHECK IF FINISHED
6485	042640	012737	063465	001330		MOV	#EM3, EM4N	:LOAD ERROR MESSAGE
6486	042646	012737	067154	001332		MOV	#EM1009, EM4N+2	
6487	042654	104004				ERROR	4	
6488	042656	104415			19\$:	SCOP1		:CHECK IF LOOP ON ERROR
6489	042660	000261				SEC		:SHIFT IN ONE
6490	042662	006137	002010			ROL	CONFIG	
6491	042666	005301				DEC	R1	:CHECK IF FINISHED
6492	042670	001402				BEQ	TST54	:YES, GO ON TO NEXT TEST
6493	042672	000137	042076			JMP	1\$	

```

6494
6495
6496 *****
6497 *TEST 54 REGISTER INTERACTION USING MAINT REG 1 (PART 3)
6498 *
6499 * ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.
6500 * WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE
6501 * REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS
6502 * ARE CORRECT AND NO INTERACTION TAKES PLACE.
6503 *
6504 * 000001 000037 000777 017777 000000
6505 * 000003 000077 001777 037777
        * 000007 000177 003777 077777
    
```



6506  
6507  
6508  
6509 042676 000004  
6510 042700 012737 000764 001200  
6511 042706 012701 000021  
6512 042712 005037 002010  
6513 042716 012737 064601 001320  
6514 042724 012762 100000 000000  
6515 042732 012737 042740 001110  
6516  
6517  
6518 042740  
6519 042740 005062 000002  
6520 042744 013762 002010 000026  
6521 042752 016237 000026 001126  
6522 042760 013737 002010 001124  
6523 042766 042737 174000 001124  
6524 042774 052737 002000 001124  
6525 043002 032737 020000 001126  
6526 043010 001403  
6527 043012 052737 020000 001124  
6528 043020  
6529 043020 023737 001124 001126  
6530 043026 001404  
6531 043030 012737 067154 001322  
6532 043036 104003  
6533 043040  
6534 043040 016237 000000 001126  
6535 043046 022737 000200 001126  
6536 043054 001407  
6537 043056 012737 000100 001124  
6538 043064 012737 067376 001322  
6539 043072 104003  
6540 043074  
6541 043074 016237 000004 001126  
6542 043102 001406  
6543 043104 005037 001124  
6544 043110 012737 067447 001322  
6545 043116 104003  
6546 043120  
6547 043120 016237 000002 001126  
6548 043126 001406  
6549 043130 005037 001124  
6550 043134 012737 067417 001322  
6551 043142 104003  
6552 043144  
6553 043144 016237 000006 001126  
6554 043152 001406  
6555 043154 005037 001124  
6556 043160 012737 067474 001322  
6557 043166 104003  
6558 043170  
6559 043170 016237 000016 001126  
6560 043176 001406  
6561 043200 005037 001124

000017 000377 007777 177777  
\*\*\*\*\*  
T54: SCOPE  
MOV #500.,\$TIMES ;:DO 500. ITERATIONS  
MOV #17.,R1 ;:LOAD NUMBER OF PATTERNS  
CLR CONFIG ;:LOAD INITIAL CONFIGURATION  
MOV #EM20,EM3N ;:LOAD ERROR MESSAGE  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$,\$LPERR ;:LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP  
  
1\$:  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG.  
MOV CONFIG,RKMR1(R2) ;:WRITE RKMR1  
MOV RKMR1(R2),\$BDDAT ;:STORE RKMR1  
MOV CONFIG,\$GDDAT ;:PREPARE EXPECTED RESULTS  
BIC #PCA!PCD!ECCW!WRTGAT!RDGATE,\$GDDAT ;:INITIALIZE READ ONLY BITS  
BIS #MEWD,\$GDDAT  
BIT #ECCW,\$BDDAT  
BEQ 30\$  
BIS #ECCW,\$GDDAT  
  
30\$:  
CMP \$GDDAT,\$BDDAT ;:CHECK IF RKMR1 CORRECT  
BEQ 2\$ ;:YES,TEST IF ANY OTHER REG MODIFIED  
MOV #EM1009,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
2\$:  
MOV RKCS1(R2),\$BDDAT ;:STORE COMMAND AND STATUS REG. 1  
CMP #RDY,\$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 3\$ ;:YES, CONTINUE  
MOV #IR,\$GDDAT ;:LOAD EXPECTED RESULTS  
MOV #EM1017,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
3\$:  
MOV RKBA(R2),\$BDDAT ;:STORE BUS AND REG  
BEQ 4\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
4\$:  
MOV RKWC(R2),\$BDDAT ;:STORE WORD COUNT REG  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
5\$:  
MOV RKDA(R2),\$BDDAT ;:STORE DISK AVERAGE REG  
BEQ 6\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1020,EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
  
6\$:  
MOV RKASOF(R2),\$BDDAT ;:STORE ATTENTION SUMMARY/OFFSET REG.  
BEQ 7\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS



6562	043204	012737	067640	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
6563	043212	104003				ERROR	3	
6564	043214				7\$:			
6565	043214	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG.2
6566	043222	022737	000100	001126		CMP	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
6567	043230	001407				BEQ	8\$	;YES, CONTINUE
6568	043232	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
6569	043240	012737	067532	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
6570	043246	104003				ERROR	3	
6571	043250				8\$:			
6572	043250	016237	000012	001126		MOV	RKDC(R2), \$BDDAT	;STORE DRIVE STATUS REG
6573	043256	001406				BEQ	9\$	;CHECK IF ZERO
6574	043260	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6575	043264	012737	067553	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
6576	043272	104003				ERROR	3	
6577	043274	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
6578	043302	001406				BEQ	10\$	;CHECK IF ZERO
6579	043304	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6580	043310	012737	067611	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
6581	043316	104003				ERROR	3	
6582	043320				10\$:			
6583	043320	016237	000020	001126		MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
6584	043326	001406				BEQ	12\$	;CHECK IF ZERO
6585	043330	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6586	043334	012737	067706	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
6587	043342	104003				ERROR	3	
6588	043344				12\$:			
6589	043344	016237	000032	001126		MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
6590	043352	001406				BEQ	15\$	;CHECK IF ZERO
6591	043354	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6592	043360	012737	070011	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
6593	043366	104003				ERROR	3	
6594	043370	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG.
6595	043376	012737	004066	001124	16\$:	MOV	#4066, \$GDDAT	;USE 4066
6596	043404	023737	001124	001126	17\$:	CMP	\$GDDAT, \$BDDAT	;CHECK IF ECC POSITION CORRECT
6597	043412	001404				BEQ	18\$	;YES, INITIALIZE RK611
6598	043414	012737	067764	001322		MOV	#EM1029,EM3N+2	;LOAD ERROR MESSAGE
6599	043422	104003				ERROR	3	
6600	043424	016237	000026	002014	18\$:	MOV	RKMR1(R2), PREREG	;GET PREVIOUS CONTENTS
6601	043432	012762	100000	000000		MOV	#CCLR, RKCS1(R2)	;CLEAR RK611 CONTROLLER
6602	043440	016237	000026	001126		MOV	RKMR1(R2), \$BDDAT	;GET CURRENT VALUE
6603	043446	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED CONTENTS
6604	043454	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6605	043462	001403				BEQ	35\$	
6606	043464	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6607	043472				35\$:			
6608	043472	023737	001124	001126		CMP	\$GDDAT, \$BDDAT	;CHECK IF RKMR1 CORRECT
6609	043500	001407				BEQ	19\$	;YES, CHECK IF FINISHED
6610	043502	012737	063465	001330		MOV	#EM3, EM4N	;LOAD ERROR MESSAGE
6611	043510	012737	067154	001332		MOV	#EM1009, EM4N+2	
6612	043516	104004				ERROR	4	
6613	043520	104415			19\$:	SCOPI		;CHECK IF LOOP ON ERROR
6614	043522	000261				SEC		;SHIFT IN ONE
6615	043524	006137	002010			ROL	CONFIG	
6616	043530	005301				DEC	R1	;CHECK IF FINISHED
6617	043532	001402				BEQ	TST55	;:YES, GO ON TO NEXT TEST



6618 043534 000137 042740

JMP 1\$

6619

6620

6621

6622

6623

6624

6625

6626

6627

6628

6629

6630

6631

6632

6633

6634 043540 000004

6635 043542 012737 000764 001200

6636 043550 012701 000021

6637 043554 005037 002010

6638 043560 012737 064601 001320

6639 043566 012762 100000 000000

6640 043574 012737 043602 001110

6641

6642

6643 043602

6644 043602 005062 000002

6645 043606 013762 002010 000026

6646 043614 016237 000026 001126

6647 043622 013737 002010 001124

6648 043630 042737 174000 001124

6649 043636 052737 002000 001124

6650 043644 032737 020000 001126

6651 043652 001403

6652 043654 052737 020000 001124

6653 043662

6654 043662 023737 001124 001126

6655 043670 001404

6656 043672 012737 067154 001322

6657 043700 104003

6658 043702

6659 043702 016237 000000 001126

6660 043710 022737 000200 001126

6661 043716 001407

6662 043720 012737 000100 001124

6663 043726 012737 067376 001322

6664 043734 104003

6665 043736

6666 043736 016237 000004 001126

6667 043744 001406

6668 043746 005037 001124

6669 043752 012737 067447 001322

6670 043760 104003

6671 043762

6672 043762 016237 000002 001126

6673 043770 001406

\*\*\*\*\*  
:TEST 55 REGISTER INTERACTION USING MAINT REG 1 (PART 4)  
\*\*\*\*\*

:ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
:WRITE THE WORD COUNT REGISTER TO ZERO. WRITE MAINTENANCE  
:REGISTER 1 WITH THE FOLLOWING CONFIGURATIONS. CHECK IF CONTENTS  
:ARE CORRECT AND NO INTERACTION TAKES PLACE.

:100000 174000 177600 177770 000000  
:140000 176000 177700 177774  
:160000 177000 177740 177776  
:170000 177400 177760 177777  
\*\*\*\*\*

TST55: SCOPE  
MOV #500, \$TIMES ;DO 500. ITERATIONS  
MOV #17, R1 ;LOAD NUMBER OF PATTERNS  
CLR CONFIG ;LOAD INITIAL CONFIGURATION  
MOV #EM20, EM3N ;LOAD ERROR MESSAGE  
MOV #CLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
MOV #1\$, \$LPERR ;LOAD LOOP ON ERROR LOCATION FOR  
; SUBTEST LOOP

1\$:  
CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MOV CONFIG, RKMR1(R2) ;WRITE RKMR1  
MOV RKMR1(R2), \$BDDAT ;STORE RKMR1  
MOV CONFIG, \$GDDAT ;PREPARE EXPECTED RESULTS  
BIC #PCA!PCD!ECCW!WRTGAT!RDGATE, \$GDDAT ;INITIALIZE READ ONLY BITS  
BIS #MEWD, \$GDDAT  
BIT #ECCW, \$BDDAT  
BEQ 30\$  
BIS #ECCW, \$GDDAT

30\$:  
CMP \$GDDAT, \$BDDAT ;CHECK IF RKMR1 CORRECT  
BEQ 2\$ ;YES, TEST IF ANY OTHER REG MODIFIED  
MOV #EM1009, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

2\$:  
MOV RKCS1(R2), \$BDDAT ;STORE COMMAND AND STATUS REG. 1  
CMP #RDY, \$BDDAT ;CHECK IF CS1 CORRECT  
BEQ 3\$ ;YES, CONTINUE  
MOV #IR, \$GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

3\$:  
MOV RKBA(R2), \$BDDAT ;STORE BUS AND REG  
BEQ 4\$ ;CHECK IF ZERO  
CLR \$GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3

4\$:  
MOV RKWC(R2), \$BDDAT ;STORE WORD COUNT REG  
BEQ 5\$ ;CHECK IF ZERO



6674	043772	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6675	043776	012737	067417	001322	MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
6676	044004	104003			ERROR	3	
6677	044006				5\$:		
6678	044006	016237	000006	001126	MOV	RKDA(R2), \$BDDAT	:STORE DISK AVERAGE REG
6679	044014	001406			BEQ	6\$	:CHECK IF ZERO
6680	044016	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6681	044022	012737	067474	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
6682	044030	104003			ERROR	3	
6683	044032				6\$:		
6684	044032	016237	000016	001126	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY/OFFSET REG.
6685	044040	001406			BEQ	7\$	:CHECK IF ZERO
6686	044042	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6687	044046	012737	067640	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
6688	044054	104003			ERROR	3	
6689	044056				7\$:		
6690	044056	016237	000010	001126	MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG.2
6691	044064	022737	000100	001126	COMP	#IR, \$BDDAT	:CHECK IF CS2 CORRECT
6692	044072	001407			BEQ	8\$	:YES, CONTINUE
6693	044074	012737	000100	001124	MOV	#IR, \$GDDAT	:LOAD EXPECTED CONTENTS
6694	044102	012737	067532	001322	MOV	#EM1021,EM3N+2	:LOAD ERROR MESSAGE
6695	044110	104003			ERROR	3	
6696	044112				8\$:		
6697	044112	016237	000012	001126	MOV	RKDS(R2), \$BDDAT	:STORE DRIVE STATUS REG
6698	044120	001406			BEQ	9\$	:CHECK IF ZERO
6699	044122	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6700	044126	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
6701	044134	104003			ERROR	3	
6702	044136	016237	000014	001126	9\$:		
6703	044144	001406			MOV	RKER(R2), \$BDDAT	:STORE ERROR REG
6704	044146	005037	001124		BEQ	10\$	:CHECK IF ZERO
6705	044152	012737	067611	001322	CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6706	044160	104003			MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
6707	044162				ERROR	3	
6708	044162	016237	000020	001126	10\$:		
6709	044170	001406			MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADD REG
6710	044172	005037	001124		BEQ	12\$	:CHECK IF ZERO
6711	044176	012737	067706	001322	CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6712	044204	104003			MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
6713	044206				ERROR	3	
6714	044206	016237	000032	001126	12\$:		
6715	044214	001406			MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
6716	044216	005037	001124		BEQ	15\$	:CHECK IF ZERO
6717	044222	012737	070011	001322	CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6718	044230	104003			MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
6719	044232	016237	000030	001126	ERROR	3	
6720	044240	012737	004066	001124	15\$:		
6721	044246	023737	001124	001126	16\$:		
6722	044254	001404			17\$:		
6723	044256	012737	067764	001322	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.
6724	044264	104003			MOV	#4066, \$GDDAT	:USE 4066
6725	044266	016237	000026	002014	18\$:		
6726	044274	012762	100000	000000	MOV	\$GDDAT, \$BDDAT	:CHECK IF ECC POSITION CORRECT
6727	044302	016237	000026	001126	BEQ	18\$	:YES, INITIALIZE RK611
6728	044310	012737	002000	001124	MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
6729	044316	032737	020000	001126	ERROR	3	
					18\$:		
					MOV	RKMR1(R2), PREREG	:GET PREVIOUS CONTENTS
					MOV	#CLR, RKCS1(R2)	:CLEAR RK611 CONTROLLER
					MOV	RKMR1(R2), \$BDDAT	:GET CURRENT VALUE
					MOV	#MEWD, \$GDDAT	:LOAD EXPECTED CONTENTS
					BIT	#ECCW, \$BDDAT	



CZR6AD0 RK611 DSKLS CTRL PRT1  
CZR6AD.P11 14-SEP-81 13:43

MACV11 30(1046)  
T55

14-SEP-81 15:04 PAGE 126  
REGISTER INTERACTION USING MAINT REG 1 (PART 4)

SEQ 0125

6730	044324	001403			BEQ	35\$	
6731	044326	052737	020000	001124	BIS	#ECCW,\$GDDAT	
6732	044334						35\$:
6733	044334	023737	001124	001126	CMP	\$GDDAT,\$BDDAT	:CHECK IF RKMRI CORRECT
6734	044342	001407			BEQ	19\$	:YES, CHECK IF FINISHED
6735	044344	012737	063465	001330	MOV	#EM3,EM4N	:LOAD ERROR MESSAGE
6736	044352	012737	067154	001332	MOV	#EM1009,EM4N+2	
6737	044360	104004			ERROR	4	
6738	044362	104415					19\$:
6739	044364	000261			SCOP1		:CHECK IF LOOP ON ERROR
6740	044366	006037	002010		SEC		:SHIFT IN ONE
6741	044372	005301			ROR	CONF'G	
6742	044374	001402			DEC	R1	:CHECK IF FINISHED
6743	044376	000137	043602		BEQ	TST56	::YES, GO ON TO NEXT TEST
6744					JMP	1\$	



6745  
6746  
6747  
6748  
6749  
6750  
6751  
6752  
6753  
6754  
6755 044402 000004  
6756 044404 012737 000764 001200  
6757 044412 013702 001270  
6758 044416 012737 177777 002010  
6759 044424 012737 064302 001320  
6760 044432 012762 100000 000000  
6761 044440 005062 000002  
6762 044444 012762 177777 000032  
6763 044452 016237 000000 001126  
6764 044460 022737 000200 001126  
6765 044466 001407  
6766 044470 012737 000200 001124  
6767 044476 012737 067376 001322  
6768 044504 104003  
6769 044506 016237 000004 001126 1\$:  
6770 044514 001406  
6771 044516 005037 001124  
6772 044522 012737 067447 001322  
6773 044530 104003  
6774 044532 016237 000002 001126 2\$:  
6775 044540 001406  
6776 044542 005037 001124  
6777 044546 012737 067417 001322  
6778 044554 104003  
6779 044556 016237 000006 001126 3\$:  
6780 044564 001406  
6781 044566 005037 001124  
6782 044572 012737 067474 001322  
6783 044600 104003  
6784 044602 016237 000016 001126 4\$:  
6785 044610 001406  
6786 044612 005037 001124  
6787 044616 012737 067640 001322  
6788 044624 104003  
6789 044626 016237 000010 001126 5\$:  
6790 044634 022737 000100 001126  
6791 044642 001407  
6792 044644 012737 000100 001124  
6793 044652 012737 067532 001322  
6794 044660 104003  
6795 044662 016237 000012 001126 6\$:  
6796 044670 001406  
6797 044672 005037 001124  
6798 044676 012737 067553 001322  
6799 044704 104003  
6800 044706 016237 000014 001126 7\$:

```
*****  
*TEST 56 REGISTER INTERACTION WITH PATTERN REG.  
*  
* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
* WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE PATTERN  
* REGISTER TO 177777 AND MAKE SURE IT REMAINS 0 AND  
* NO INTERACTION TAKES PLACE.  
*****  
TST56: SCOPE  
MOV #500, $TIMES ;DO 500. ITERATIONS  
MOV $BASE, R2 ;LOAD RK611 BASE ADDRESS  
MOV #177777, CONFIG ;LOAD CONFIGURATION WORD  
MOV #EM14, EM3N ;LOAD ERROR MESSAGE  
MOV #CCLR, RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR  
CLR RKWC(R2) ;CLEAR WORD COUNT REG.  
MCV #177777, RKECPT(R2) ;WRITE RKECPT WITH 177777  
MOV RKCS1(R2), $BDDAT ;STORE COMMAND AND STATUS REG.1  
CMP #RDY, $BDDAT ;CHECK IF CS1 CORRECT  
BEQ 1$ ;YES, CONTINUE  
MOV #RDY, $GDDAT ;LOAD EXPECTED RESULTS  
MOV #EM1017, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKBA(R2), $BDDAT ;STORE BUS ADD REG.  
BEQ 2$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKWC(R2), $BDDAT ;STORE WORK COUNT REG.  
BEQ 3$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKDA(R2), $BDDAT ;STORE DISK ADD REG  
BEQ 4$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1020, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKASOF(R2), $BDDAT ;STORE ATTENTION SUMMARY/OFFSET REG  
BEQ 5$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1024, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKCS2(R2), $BDDAT ;STORE COMMAND AND STATUS REG. 2  
CMP #IR, $BDDAT ;CHECK IF CS2 CORRECT  
BEQ 6$ ;YES, CONTINUE  
MOV #IR, $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1021, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKDS(R2), $BDDAT ;STORE DRIVE STATUS REG.  
BEQ 7$ ;CHECK IF ZERO  
CLR $GDDAT ;LOAD EXPECTED CONTENTS  
MOV #EM1022, EM3N+2 ;LOAD ERROR MESSAGE  
ERROR 3  
MOV RKER(R2), $BDDAT ;STORE ERROR REG
```



6801	044714	001406				BEQ	8\$	:CHECK IF ZERO
6802	044716	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6803	044722	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
6804	044730	104003				ERROR	3	
6805	044732	016237	000020	001126	8\$:	MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADD REG
6806	044740	001406				BEQ	10\$	:CHECK IF ZERO
6807	044742	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED RESULTS
6808	044746	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
6809	044754	104003				ERROR	3	
6810	044756	016237	000026	001126	10\$:	MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG 1
6811	044764	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED CONTENTS
6812	044772	032737	020000	001126		BIT	#ECCW,\$BDDAT	
6813	045000	001403				BEQ	11\$	
6814	045002	052737	020000	001124		BIS	#ECCW,\$GDDAT	
6815	045010	023737	001124	001126	11\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
6816	045016	001404				BEQ	12\$	:YES,CONTINUE TEST
6817	045020	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
6818	045026	104003				ERROR	3	
6819	045030	016237	000032	001126	12\$:	MOV	RKECPT(R2),\$BDDAT	:STORE ECC PATTERN REG.
6820	045036	001406				BEQ	13\$	:CHECK IF ZERO
6821	045040	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
6822	045044	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
6823	045052	104003				ERROR	3	
6824	045054	016237	000030	001126	13\$:	MOV	RKECPS(R2),\$BDDAT	:STORE ECC POSITION REG:
6825	045062	022737	004066	001126		CMP	#4066,\$BDDAT	:CHECK IF ECC POSITION CORRECT
6826	045070	001407				BEQ	TST57	::YES,GO TO NEXT TEST
6827	045072	012737	004066	001124		MOV	#4066,\$GDDAT	:LOAD EXPECTED CONTENTS
6828	045100	012737	067764	001322		MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
6829	045106	104003				ERROR	3	

```

6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856

```

\*\*\*\*\*  
\*TEST 57 REGISTER INTERACTION WITH POSITION REG.  
\*  
\* ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 REGISTERS.  
\* WRITE THE WORD COUNT WITH REGISTER TO ZERO. WRITE POSITION  
\* REGISTER TO 177777 AND MAKE SURE IT STAYS AT THE  
\* INITIALIZED CONDITION AND NO INTERACTION TAKES PLACE.  
\*\*\*\*\*

6840	045110	000004				TST57:	SCOPE	
6841	045112	012737	000764	001200		MOV	#500,\$TIMES	::DO 500. ITERATIONS
6842	045120	013702	001270			MOV	\$BASE,R2	:LOAD RK611 BASE ADDRESS
6843	045124	012737	177777	002010		MOV	#177777,CONFIG	:LOAD CONFIGURATION WORD
6844	045132	012737	064342	001320		MOV	#EM15,EM3N	:LOAD ERROR MESSAGE
6845	045140	012762	100000	000000		MOV	#CLR,RKCS1(R2)	:CLEAR RK611 WITH CONTROLLER CLEAR
6846	045146	005062	000002			CLR	RKWC(R2)	:CLEAR WORD COUNT REG.
6847	045152	012762	177777	000030		MOV	#177777,RKECPS(R2)	:WRITE RKECPS WITH 177777
6848	045160	016237	000000	001126		MOV	RKCS1(R2),\$BDDAT	:STORE COMMAND AND STATUS REG.1
6849	045166	022737	000200	001126		CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
6850	045174	001407				BEQ	1\$	:YES,CONTINUE
6851	045176	012737	000200	001124		MOV	#RDY,\$GDDAT	:LOAD EXPECTED RESULTS
6852	045204	012737	067376	001322		MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
6853	045212	104003				ERROR	3	
6854	045214	016237	000004	001126	1\$:	MOV	RKBA(R2),\$BDDAT	:STORE BUS ADD REG.
6855	045222	001406				BEQ	2\$	:CHECK IF ZERO
6856	045224	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS



6857	045230	012737	067447	001322		MOV	#EM1019,EM3N+2	;LOAD ERROR MESSAGE
6858	045236	104003				ERROR	3	
6859	045240	016237	000002	001126	2\$:	MOV	RKWC(R2), \$BDDAT	;STORE WORK COUNT REG.
6860	045246	001406				BEQ	3\$	;CHECK IF ZERO
6861	045250	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6862	045254	012737	067417	001322		MOV	#EM1018,EM3N+2	;LOAD ERROR MESSAGE
6863	045262	104003				ERROR	3	
6864	045264	016237	000006	001126	3\$:	MOV	RKDA(R2), \$BDDAT	;STORE DISK ADD REG
6865	045272	001406				BEQ	4\$	;CHECK IF ZERO
6866	045274	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6867	045300	012737	067474	001322		MOV	#EM1020,EM3N+2	;LOAD ERROR MESSAGE
6868	045306	104003				ERROR	3	
6869	045310	016237	000016	001126	4\$:	MOV	RKASOF(R2), \$BDDAT	;STORE ATTENTION SUMMARY/OFFSET REG
6870	045316	001406				BEQ	5\$	;CHECK IF ZERO
6871	045320	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6872	045324	012737	067640	001322		MOV	#EM1024,EM3N+2	;LOAD ERROR MESSAGE
6873	045332	104003				ERROR	3	
6874	045334	016237	000010	001126	5\$:	MOV	RKCS2(R2), \$BDDAT	;STORE COMMAND AND STATUS REG. 2
6875	045342	022737	000100	001126		CMF	#IR, \$BDDAT	;CHECK IF CS2 CORRECT
6876	045350	001407				BEQ	6\$	;YES, CONTINUE
6877	045352	012737	000100	001124		MOV	#IR, \$GDDAT	;LOAD EXPECTED CONTENTS
6878	045360	012737	067532	001322		MOV	#EM1021,EM3N+2	;LOAD ERROR MESSAGE
6879	045366	104003				ERROR	3	
6880	045370	016237	000012	001126	6\$:	MOV	RKDS(R2), \$BDDAT	;STORE DRIVE STATUS REG.
6881	045376	001406				BEQ	7\$	;CHECK IF ZERO
6882	045400	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6883	045404	012737	067553	001322		MOV	#EM1022,EM3N+2	;LOAD ERROR MESSAGE
6884	045412	104003				ERROR	3	
6885	045414	016237	000014	001126	7\$:	MOV	RKER(R2), \$BDDAT	;STORE ERROR REG
6886	045422	001406				BEQ	8\$	;CHECK IF ZERO
6887	045424	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6888	045430	012737	067611	001322		MOV	#EM1023,EM3N+2	;LOAD ERROR MESSAGE
6889	045436	104003				ERROR	3	
6890	045440	016237	000020	001126	8\$:	MOV	RKDCYL(R2), \$BDDAT	;STORE CYLINDER ADD REG
6891	045446	001406				BEQ	10\$	;CHECK IF ZERO
6892	045450	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED RESULTS
6893	045454	012737	067706	001322		MOV	#EM1025,EM3N+2	;LOAD ERROR MESSAGE
6894	045462	104003				ERROR	3	
6895	045464	016237	000026	001126	10\$:	MOV	RKMR1(R2), \$BDDAT	;STORE MAINTENANCE REG 1
6896	045472	012737	002000	001124		MOV	#MEWD, \$GDDAT	;LOAD EXPECTED CONTENTS
6897	045500	032737	020000	001126		BIT	#ECCW, \$BDDAT	
6898	045506	001403				BEQ	11\$	
6899	045510	052737	020000	001124		BIS	#ECCW, \$GDDAT	
6900	045516	023737	001124	001126	11\$:	CMF	\$GDDAT, \$BDDAT	;CHECK IF MR1 CORRECT
6901	045524	001404				BEQ	12\$	;YES, CONTINUE TEST
6902	045526	012737	067733	001322		MOV	#EM1026,EM3N+2	;LOAD ERROR MESSAGE
6903	045534	104003				ERROR	3	
6904	045536	016237	000032	001126	12\$:	MOV	RKECPT(R2), \$BDDAT	;STORE ECC PATTERN REG.
6905	045544	001406				BEQ	13\$	;CHECK IF ZERO
6906	045546	005037	001124			CLR	\$GDDAT	;LOAD EXPECTED CONTENTS
6907	045552	012737	070011	001322		MOV	#EM1030,EM3N+2	;LOAD ERROR MESSAGE
6908	045560	104003				ERROR	3	
6909	045562	016237	000030	001126	13\$:	MOV	RKECPS(R2), \$BDDAT	;STORE ECC POSITION REG:
6910	045570	022737	004066	001126		CMF	#4066, \$BDDAT	;CHECK IF ECC POSITION CORRECT
6911	045576	001407				BEQ	TST60	:::YES, GO TO NEXT TEST
6912	045600	012737	004066	001124		MOV	#4066, \$GDDAT	;LOAD EXPECTED CONTENTS



6913 045606 012737 067764 001322  
 6914 045614 104003  
 6915  
 6916  
 6917  
 6918  
 6919  
 6920  
 6921  
 6922  
 6923  
 6924  
 6925  
 6926  
 6927  
 6928  
 6929  
 6930 045616 000004  
 6931 045620 012737 000764 001200  
 6932 045626 012701 070036  
 6933 045632 005000  
 6934 045634 012703 000400  
 6935 045640 012021  
 6936 045642 005303  
 6937 045644 001375  
 6938 045646 012737 177777 002016  
 6939 045654 005000  
 6940 045656 012701 046010  
 6941 045662 012703 000200  
 6942 045666 010120  
 6943 045670 012720 000340  
 6944 045674 062701 000002  
 6945 045700 005303  
 6946 045702 001371  
 6947 045704 012762 100000 000000  
 6948 045712 012705 046410  
 6949 045716 005046  
 6950 045720 012746 045726  
 6951 045724 000002  
 6952  
 6953 045726  
 6954 045726 000240  
 6955 045730 012705 046462  
 6956 045734 012762 000300 000000  
 6957 045742 000240  
 6958 045744 012746 000340  
 6959 045750 012746 045756  
 6960 045754 000002  
 6961  
 6962 045756 012701 070036  
 6963 045762 005000  
 6964 045764 012703 000400  
 6965 045770 012120  
 6966 045772 005303  
 6967 045774 001375  
 5968 045776 005037 002016

```

MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
ERROR 3

.SBTTL **INTERRUPT TESTS
*****
*TEST 60 RK611 INTERRUPT
*****
* STORE LOCATIONS 0-776, LOAD LOCATIONS 0-776 TO TRAP ALL
* POSSIBLE INTERRUPTS. LOWER PROCESSOR PRIORITY TO ZERO.
* MAKE SURE THAT NO INTERRUPT OCCURS. NOW SET INTERRUPT
* ENABLE AND READY. VERIFY THAT THE INTERRUPT OCCURS AT
* PROPER VECTOR ADDRESS. MAKE SURE THAT INTERRUPT IS
* CLEARED AFTER IT IS GIVEN.
*****
TST60: SCOPE
MOV #500,$TIMES ;DO 500. ITERATIONS
MOV #SAVVEC,R1 ;LOAD SAVED ADDRESS
CLR R0 ;LOAD START OF VECTOR SPACE
MOV #400,R3 ;LOAD COUNT
1$: MOV (R0)+,(R1)+ ;SAVE TRAP CATCHER
DEC R3
BNE 1$
MOV #-1,SAVFLG ;INDICATE TRAP CATCHER SAVED
CLR R0 ;LOAD ADDRESS OF START OF VECTOR SPACE
MOV #10$,R1 ;LOAD START OF VECTOR TABLE
MOV #200,R3 ;LOAD COUNT
2$: MOV R1,(R0)+ ;LOAD VECTOR SPACE
MOV #PR7,(R0)+ ; PRIORITY 7
ADD #2,R1
DEC R3
BNE 2$
MOV #CCLR,RKCS1(R2) ;CLEAR RK611 WITH CONTROLLER CLEAR
MOV #11$,R5 ;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
CLR -(SP) ;LOAD STACK TO ALLOW ALL INTERRUPTS
MOV #64$,-(SP) ;LOAD NEXT ADDRESS
RTI ;CLEAR PSW

64$:
NOP ;ALLOW INTERRUPT TO OCCUR
MOV #15$,R5 ;LOAD ADDRESS FOR EXPECTED INTERRUPT
MOV #RDY!IE,RKCS1(R2) ;GENERATE INTERRUPT
NOP ;ALLOW INTERRUPT TO OCCUR
MOV #PR7,-(SP) ;LOCK OUT ALL INTERRUPTS
MOV #3$,-(SP)

3$: MOV #SAVVEC,R1 ;LOAD SAVED TRAP CATCHER ADDRESS
CLR R0 ;LOAD START OF VECTOR SPACE
MOV #400,R3 ;LOAD COUNT
4$: MOV (R1)+,(R0)+ ;RESTORE TRAP CATCHER
DEC R3
BNE 4$
CLR SAVFLG ;INDICATE TRAP CATCHER RESTORED
  
```



6969	046002	104005		ERROR	5		:REPORT INTERRUPT DID NOT OCCUR
6970	046004	000137	046672	JMP	60\$		:GO ON TO NEXT TEST
6971							
6972	046010						
6973	046010	004715	10\$:	JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6974	046012	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6975	046014	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6976	046016	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6977	046020	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6978	046022	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6979	046024	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6980	046026	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6981	046030	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6982	046032	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6983	046034	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6984	046036	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6985	046040	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6986	046042	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6987	046044	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6988	046046	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6989	046050	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6990	046052	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6991	046054	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6992	046056	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6993	046060	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6994	046062	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6995	046064	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6996	046066	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6997	046070	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6998	046072	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
6999	046074	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7000	046076	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7001	046100	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7002	046102	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7003	046104	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7004	046106	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7005	046110	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7006	046112	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7007	046114	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7008	046116	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7009	046120	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7010	046122	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7011	046124	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7012	046126	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7013	046130	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7014	046132	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7015	046134	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7016	046136	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7017	046140	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7018	046142	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7019	046144	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7020	046146	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7021	046150	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7022	046152	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7023	046154	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS
7024	046156	004715		JSR	PC,(R5)		:LOAD STACK FOR VECTOR ADDRESS



7025	046160	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7026	046162	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7027	046164	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7028	046166	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7029	046170	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7030	046172	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7031	046174	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7032	046176	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7033	046200	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7034	046202	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7035	046204	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7036	046206	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7037	046210	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7038	046212	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7039	046214	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7040	046216	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7041	046220	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7042	046222	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7043	046224	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7044	046226	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7045	046230	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7046	046232	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7047	046234	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7048	046236	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7049	046240	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7050	046242	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7051	046244	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7052	046246	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7053	046250	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7054	046252	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7055	046254	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7056	046256	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7057	046260	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7058	046262	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7059	046264	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7060	046266	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7061	046270	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7062	046272	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7063	046274	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7064	046276	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7065	046300	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7066	046302	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7067	046304	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7068	046306	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7069	046310	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7070	046312	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7071	046314	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7072	046316	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7073	046320	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7074	046322	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7075	046324	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7076	046326	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7077	046330	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7078	046332	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7079	046334	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7080	046336	004715	JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS



7081	046340	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7082	046342	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7083	046344	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7084	046346	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7085	046350	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7086	046352	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7087	046354	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7088	046356	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7089	046360	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7090	046362	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7091	046364	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7092	046366	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7093	046370	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7094	046372	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7095	046374	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7096	046376	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7097	046400	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7098	046402	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7099	046404	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7100	046406	004715				JSR	PC,(R5)	;LOAD STACK FOR VECTOR ADDRESS
7101								
7102	046410	012637	001122		11\$:	MOV	(SP)+,\$BDADR	;DETERMINE VECTOR ADDRESS
7103	046414	162737	046012	001122		SUB	#10\$+2,\$BDADR	
7104	046422	006337	001122			ASL	\$BDADR	
7105	046426	062706	000004			ADD	#4,SP	;ADJUST STACK
7106	046432	012701	070036			MOV	#SAVVEC,R1	;LOAD SAVED TRAP CATCHER ADDRESS
7107	046436	005000				CLR	R0	;LOAD START OF VECTOR SPACE
7108	046440	012703	000400			MOV	#400,R3	;LOAD COUNT
7109	046444	012120			12\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
7110	046446	005303				DEC	R3	
7111	046450	001375				BNE	12\$	
7112	046452	005037	002016			CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
7113	046456	104006				ERROR	6	;REPORT UNEXPECTED INTERRUPT
7114	046460	000504				BR	TST61	;GO ON TO NEXT TEST
7115								
7116	046462	012637	001122		15\$:	MOV	(SP)+,\$BDADR	;DETERMINE VECTOR ADDRESS
7117	046466	162737	046012	001122		SUB	#10\$+2,\$BDADR	
7118	046474	006337	001122			ASL	\$BDADR	
7119	046500	062706	000004			ADD	#4,SP	;ADJUST STACK
7120	046504	023737	002000	001122		CMP	RKVEC,\$BDADR	;CHECK IF VECTOR ADDRESS CORRECT
7121	046512	001414				BEQ	20\$	;YES, CONTINUE
7122	046514	012701	070036			MOV	#SAVVEC,R1	;LOAD SAVED TRAP CATCHER ADDRESS
7123	046520	005000				CLR	R0	;START OF VECTOR SPACE
7124	046522	012703	000400			MOV	#400,R3	;LOAD COUNT
7125	046526	012120			16\$:	MOV	(R1)+,(R0)+	;RESTORE TRAP CATCHER
7126	046530	005303				DEC	R3	
7127	046532	001375				BNE	16\$	
7128	046534	005037	002016			CLR	SAVFLG	;INDICATE TRAP CATCHER RESTORED
7129	046540	104007				ERROR	7	;REPORT INCORRECT VECTOR ADDRESS
7130	046542	000453				BR	TST61	;GO ON TO NEXT TEST
7131								
7132	046544	012705	046622		20\$:	MOV	#25\$,R5	;LOAD ADDRESS FOR UNEXPECTED INTERRUPT
7133	046550	005046				CLR	-(SP)	;LOAD STACK TO ALLOW ALL INTERRUPTS
7134	046552	012746	046560			MOV	#65\$,-(SP)	;LOAD NEXT ADDRESS
7135	046556	000002				RTI		;CLEAR PSW
7136								



```
7137 046560 65$:
7138 046560 012746 000340 MOV #PR7,-(SP) ;LOCK OUT RK611 INTERRUPTS
7139
7140 046564 012746 046572 MOV #21$,-(SP)
7141 046570 000002 RTI
7142
7143 046572 000240 21$: NOP ;ALLOW INTERRUPT TO OCCUR
7144 046574 012701 070036 MOV #SAVVEC,R1 ;LOAD SAVE TRAP CATCHER ADDRESS
7145 046600 005000 CLR R0 ;START OF VECTOR SPACE
7146 046602 012703 000400 MOV #400,R3 ;LOAD COUNT
7147 046606 012120 22$: MOV (R1)+,(R0)+ ;RESTORE TRAP CATCHER
7148 046610 005303 DEC R3
7149 046612 001375 BNE 22$
7150 046614 005037 002016 CLR SAVFLG ;INDICATE TRAP CATCHER RESTORED
7151 046620 000424 BR TST61 ;GO ON TO NEXT TEST
7152
7153 046622 012637 001122 25$: MOV (SP)+,$BDADR ;DETERMINE VECTOR ADDRESS
7154 046626 162737 046012 001122 SUB #10$+2,$BDADR
7155 046634 006337 001122 ASL $BDADR
7156 046640 062706 000004 ADD #4,SP ;ADJUST STACK
7157 046644 012701 070036 MOV #SAVVEC,R1 ;LOAD SAVE TRAP CATCHER ADDRESS
7158 046650 005000 CLR R0 ;START OF VECTOR SPACE
7159 046652 012703 000400 MOV #400,R3 ;LOAD COUNT
7160 046656 012120 26$: MOV (R1)+,(R0)+ ;RESTORE TRAP CATCHER
7161 046660 005303 DEC R3
7162 046662 001375 BNE 26$
7163 046664 005037 002016 CLR SAVFLG ;INDICATE THAT TRAP CATCHER RESTART
7164
7165 046670 104010 ERROR 10 ;REPORT ATTENTION DID NOT CLEAR
7166 046672 60$:
7167
7168 *****
7169 :*TEST 61 INTERRUPT PRIORITY
7170 :*
7171 :* SET UP PRIORITY TO 1 LESS THAN INTERRUPT PRIORITY.
7172 :* WRITE READY WITH INTERRUPT ENABLE. MAKE SURE INTERRUPT OCCURS.
7173 :*
7174 :* NOW SET UP PRIORITY EQUAL TO INTERRUPT PRIORITY.
7175 :* WRITE INTERRUPT ENABLE WITH READY. MAKE SURE INTERRUPT
7176 :* DOES NOT OCCUR. NOW LOWER PRIORITY AND MAKE
7177 :* INTERRUPT HAS BEEN STORED.
7178 :*
7179 *****
7180 046672 000004 TST61: SCOPE
7181 046674 012737 000764 001200 MOV #500,$TIMES ;DO 500. ITERATIONS
7182 046702 013701 002000 MOV RKVEC,R1 ;LOAD RK611 VECTOR ADDRESS FOR INTERRUPT
7183 046706 012721 046776 MOV #10$,(R1)+ ; PRIORITY 7
7184 046712 013746 002002 MOV RKPRI,-(SP) ;SET PROCESSOR PRIORITY =
7185 046716 162716 00C040 SUB #40,(SP) ; RK611 PRIORITY -1
7186 046722 011646 MOV (SP),-(SP)
7187 046724 006216 ASR (SP)
7188 046726 006216 ASR (SP)
7189 046730 006216 ASR (SP)
7190 046732 006216 ASR (SP)
7191 046734 006216 ASR (SP)
7192 046736 012637 002020 MOV (SP)+,PRIOR
```



```

7193 046742 012746 046750      MOV      #1$,-(SP)
7194 046746 000002      RTI
7195
7196 046750 012762 000300 000000 1$:  MOV      #RDY!IE,RKCS1(R2)      ;GENERATE RK611 INTERRUPT
7197 046756 000240      NOP      ;WAIT FOR INTERRUPT
7198 046760 012746 000340      MOV      #PR7,-(SP)      ;LOCK OUT INTERRUPTS
7199 046764 012746 046772      MOV      #2$,-(SP)
7200 046770 000002      RTI
7201
7202 046772 104011      2$:  ERROR  11      ;EXPECTED INTERRUPT DID NOT OCCUR AT
7203                                     ; PROCESSOR PRIORITY
7204 046774 000462      BR       60$      ;RESTORE TRAP CATCHER
7205
7206 046776 062706 000004 10$:  ADD      #4,SP      ;ADJUST STACK
7207 047002 012777 047126 132770  MOV      #20$,@RKVEC      ;LOAD RK611 VECTOR FOR UNEXPECTED
7208                                     ; INTERRUPT
7209 047010 013737 002002 002020  MOV      RKPRI,PRIOR      ;STORE PRIORITY PRINT OUT
7210 047016 006237 002020      ASR     PRIOR
7211 047022 006237 002020      ASR     PRIOR
7212 047026 006237 002020      ASR     PRIOR
7213 047032 006237 002020      ASR     PRIOR
7214 047036 006237 002020      ASR     PRIOR
7215 047042 013746 002002      MOV      RKPRI,-(SP)      ;SET PROCESSOR PRIORITY
7216 047046 012746 047054      MOV      #11$,-(SP)      ; RK611 PRIORITY
7217 047052 000002      RTI
7218
7219 047054 012762 000300 000000 11$:  MOV      #RDY!IE,RKCS1(R2)      ;GENERATE RK611 INTERRUPT
7220 047062 000240      NOP      ;ALLOW INTERRUPT TO OCCUR
7221 047064 005037 002020      CLR     PRIOR      ;LOAD PRIORITY FOR PRINT OUT
7222 047070 012777 047136 132702  MOV      #25$,@RKVEC      ;LOAD RK611 VECTOR FOR INTERRUPT
7223 047076 005046      CLR     -(SP)      ;LOAD STACK TO ALLOW ALL INTERRUPTS
7224 047100 012746 047106      MOV      #64$,-(SP)      ;LOAD NEXT ADDRESS
7225 047104 000002      RTI      ;CLEAR PSW
7226
7227 047106      64$:
7228 047106 000240      NOP
7229 047110 012746 000340      MOV      #PR7,-(SP)      ;ALLOW INTERRUPT TO OCCUR
7230 047114 012746 047122      MOV      #12$,-(SP)      ;LOCK OUT INTERRUPTS
7231 047120 000002      RTI
7232
7233 047122 104013      12$:  ERROR  13      ;INTERRUPT DID NOT OCCUR WHEN
7234                                     ; PRIORITY LOWERED
7235 047124 000406      BR       60$      ;RESTORE TRAP CATCHER
7236
7237 047126 062706 000004 20$:  ADD      #4,SP      ;ADJUST STACK
7238 047132 104012      ERROR  12      ;UNEXPECTED INTERRUPT OCCURRED
7239                                     ; AT PROCESSOR PRIORITY
7240 047134 000402      BR       60$      ;RESTORE TRAP CATCHER
7241
7242 047136 062706 000004 25$:  ADD      #4,SP      ;ADJUST STACK
7243
7244 047142 013701 002000 60$:  MOV      RKVEC,R1      ;RESTORE TRAP CATCHER
7245 047146 010111      MOV      R1,(R1)
7246 047150 062721 000002      ADD      #2,(R1)+
7247 047154 005011      CLR     (R1)
7248
  
```



7249  
7250  
7251  
7252  
7253  
7254  
7255  
7256  
7257 047156 000004  
7258 047160 012737 000764 001200  
7259 047166 012762 100000 000000  
7260 047174 013701 002000  
7261 047200 012721 047242  
7262 047204 012711 000340  
7263 047210 005046  
7264 047212 012746 047220  
7265 047216 000002  
7266  
7267 047220  
7268 047220 012762 000100 000000  
7269 047226 000240  
7270 047230 012746 000340  
7271 047234 012746 047250  
7272 047240 000002  
7273  
7274 047242 062706 000004  
7275 047246 104014  
7276 047250 013701 002000  
7277 047254 010111  
7278 047256 012721 000002  
7279 047262 005011  
7280  
7281  
7282  
7283  
7284  
7285  
7286  
7287  
7288  
7289  
7290 047264 000004  
7291 047266 012737 000764 001200  
7292 047274 012762 100000 000000  
7293 047302 012762 000300 000000  
7294 047310 012762 100000 000000  
7295 047316 012762 000100 000000  
7296 047324 013701 002000  
7297 047330 012721 047364  
7298 047334 012711 000340  
7299 047340 005046  
7300 047342 012746 047350  
7301 047346 000002  
7302  
7303 047350  
7304 047350 000240

```
*****  
:TEST 62 SETTING INTERRUPT ENABLE  
:  
: CLEAR RK611 CONTROLLER WITH CONTROLLER CLEAR. ALLOW RK611  
: INTERRUPTS BY SETTING PROCESSOR PRIORITY TO ZERO.  
: SET INTERRUPT ENABLE AND MAKE SURE NO INTERRUPTS OCCUR.  
:  
*****  
TST62: SCOPE  
MOV #500, $TIMES ;:DO 500. ITERATIONS  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 CONTROLLER  
MOV RKVEC,R1 ;:LOAD RK611 VECTOR ADDRESS FOR  
MOV #10$(R1)+ ; UNEXPECTED INTERRUPT  
MOV #PR7,(R1) ; PRIORITY 7  
CLR -(SP) ;:LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV #64$,-(SP) ;:LOAD NEXT ADDRESS  
RTI ;:CLEAR PSW  
  
64$:  
MOV #IE,RKCS1(R2) ;:SET INTERRUPT ENABLE  
NOP ;:ALLOW INTERRUPT TO OCCUR  
MOV #PR7,-(SP) ;:LOCK OUT ALL INTERRUPTS  
MOV #15$,-(SP) ;:RESTORE TRAP CATCHER  
RTI  
  
10$: ADD #4,SP ;:ADJUST STACK  
ERROR 14 ;:REPORT ERROR  
  
15$: MOV RKVEC,R1 ;:RESTORE TRAP CATCHER  
MOV R1,(R1)  
MOV #2,(R1)+  
CLR (R1)
```

```
*****  
:TEST 63 INTERRUPT CLEARING  
:  
: SET UP PRIORITY TO SEVEN. CREATE INTERRUPT BY SETTING  
: INTERRUPT ENABLE READY. AND CLEAR IT WITH CONTROLLER  
: CLEAR. SET INTERRUPT ENABLE. NOW LOWER PRIORITY  
: TO MAKE SURE NO INTERRUPT OCCURS.  
:  
*****  
TST63: SCOPE  
MOV #500, $TIMES ;:DO 500. ITERATIONS  
MOV #CCLR,RKCS1(R2) ;:CLEAR RK611  
MOV #IE!RDY,RKCS1(R2) ;:GENERATE INTERRUPT  
MOV #CCLR,RKCS1(R2) ;:CLEAR INTERRUPT  
MOV #IE,RKCS1(R2) ;:GET INTERRUPT ENABLE  
MOV RKVEC,R1 ;:LOAD RK611 VECTOR ADDRESS FOR  
MOV #10$(R1)+ ; UNEXPECTED INTERRUPT  
MOV #PR7,(R1) ; PRIORITY 7  
CLR -(SP) ;:LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV #64$,-(SP) ;:LOAD NEXT ADDRESS  
RTI ;:CLEAR PSW  
  
64$:  
NOP ;:ALLOW INTERRUPT TO OCCUR
```



```
7305 047352 012746 000340      MOV    #PR7,-(SP)      ;LOCK OUT INTERRUPTS
7306 047356 012746 047372      MOV    #15$,-(SP)     ;RESTORE TRAP CATCHER
7307 047362 000002
7308
7309 047364 062706 000004      10$:  ADD    #4,SP      ;ADJUST STACK
7310 047370 104015                ERROR  15             ;REPORT ERROR
7311 047372 013701 002000      15$:  MOV    RKVEC,R1    ;RESTORE TRAP CATCHER
7312 047376 010111                MOV    R1,(R1)
7313 047400 062721 000002      ADD    #2,(R1)+
7314 047404 005011                CLR    (R1)
```

.SBTTL \*\*SILO TESTS

```
*****
*TEST 64      READ SILO WHEN EMP!Y
*
*      READ SILO WHEN EMPTY. CHECK FOR DATA LATE AND CONTROLLER
*      ERROR. ISSUE CONTROLLER CLEAR AND CHECK IF ERROR RESET.
*****
```

```
7325
7326 047406 000004
7327 047410 012737 000764 001200  TST64: SCOPE
7328 047416 013702 001270      MOV    #500,$TIMES    ;;DO 500. ITERATIONS
7329 047422 012762 100000 000000      MOV    $BASE,R2      ;LOAD RK611 BASE
7330 047430 005762 000024      MOV    #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER
7331 047434 016237 000010 001710      TST    RKDB(R2)      ;READ DATA BUFFER
7332 047442 016237 000000 001700      MOV    RKCS2(R2),T.CS2;STORE COMMAND AND STATUS REG.2
7333 047450 012737 100100 001750      MOV    RKCS1(R2),T.CS1;STORE COMMAND AND STATUS REG. 1
7334 047456 012737 100200 001740      MOV    #DLT!IR,E.CS2 ;LOAD EXPECTED CS2
7335 047464 023737 001750 001710      MOV    #CERR!RDY,E.CS1;LOAD EXPECTED CS1
7336 047472 001401                CMP    E.CS2,T.CS2   ;CHECK FOR DAT LATE SET
7337 047474 104016                BEQ    1$            ;YES, CHECK FOR CONTROLLER ERROR
7338                                ERROR  16            ;CS2 INCORRECT AFTER READING
7339                                ; EMPTY SILO
7339 047476 023737 001740 001700  1$:  CMP    E.CS1,T.CS1   ;CHECK FOR CONTROLLER ERROR
7340 047504 001401                BEQ    2$            ;YES, CLEAR DATA LATE
7341 047506 104017                ERROR  17            ;CS1 INCORRECT AFTER REQUIRE
7342                                ; EMPTY SILO
7343 047510 012762 100000 000000  2$:  MOV    #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER
7344 047516 016237 000000 001700      MOV    RKCS1(R2),T.CS1;STORE COMMAND AND STATUS REG. 1
7345 047524 016237 000010 001710      MOV    RKCS2(R2),T.CS2;STORE COMMAND AND STATUS REG.2
7346 047532 012737 000200 001740      MOV    #RDY,E.CS1    ;LOAD EXPECTED CS1
7347 047540 012737 000100 001750      MOV    #IR,E.CS2     ;LOAD EXPECTED CS2
7348 047546 023737 001740 001700      CMP    E.CS1,T.CS1   ;CHECK TO CONTROLLER ERROR RESET
7349 047554 001401                BEQ    3$            ;YES, CHECK IF DATA LATE RESET
7350 047556 104020                ERROR  20            ;CS1 INCORRECT AFTER ATTEMPTING
7351                                ; TO CLEAR DATA LATE
7352 047560 023737 001750 001710  3$:  CMP    E.CS2,T.CS2   ;CHECK IF DATA LATE RESET
7353 047566 001401                BEQ    TST65         ;YES, GO ON TO NEXT TEST
7354 047570 104021                ERROR  21            ;CS2 INCORRECT AFTER ATTEMPTING
7355                                ; TO OCCUR DATA LATE
```

```
*****
*TEST 65      SILO LOADING AND UNLOADING OF ONE WORD
*
*      ISSUE A CONTROLLER CLEAR TO INITIALIZE RK611 CONTROLLER.
*****
```

7360



7361  
7362  
7363  
7364  
7365  
7366  
7367  
7368  
7369  
7370  
7371  
7372  
7373  
7374  
7375 047572 000004  
7376 047574 012737 000764 001200  
7377 047602 012762 100000 000000  
7378 047610 005062 000002  
7379 047614 012737 177777 002010  
7380 047622 012762 177777 000024  
7381 047630 016237 000000 001126  
7382 047636 022737 000200 001126  
7383 047644 001407  
7384 047646 012737 000200 001124  
7385 047654 012737 067376 001322  
7386 047662 104003  
7387 047664 016237 000004 001126 1\$:  
7388 047672 001406  
7389 047674 005037 001124  
7390 047700 012737 067447 001322  
7391 047706 104003  
7392 047710 016237 000002 001126 2\$:  
7393 047716 001406  
7394 047720 005037 001124  
7395 047724 012737 067417 001322  
7396 047732 104003  
7397 047734 016237 000006 001126 3\$:  
7398 047742 001406  
7399 047744 005037 001124  
7400 047750 012737 067474 001322  
7401 047756 104003  
7402 047760 016237 000016 001126 4\$:  
7403 047766 001406  
7404 047770 005037 001124  
7405 047774 012737 067764 001322  
7406 050002 104003  
7407 050004 012700 000005 5\$:  
7408 050010 005300 6\$:  
7409 050012 001376  
7410 050014 016237 000010 001126  
7411 050022 022737 000300 001126  
7412 050030 001407  
7413 050032 012737 000300 001124  
7414 050040 012737 066766 001322  
7415 050046 104003  
7416 050050 016237 000012 001126 7\$:

\*\*\*  
: CLEAR WORD COUNT REGISTER.  
: WRITE A WORD OF 177777 INTO THE SILO. CHECK ALL OTHER  
: REGISTERS FOR INTERACTION PROBLEMS. CHECK THAT OUTPUT  
: READY IS SET IN COMMAND AND STATUS REGISTER 2. IF NOT  
: WAIT A REASONABLE TIME FOR IT.  
: IF OUTPUT READY COMES UP IN A REASONABLE TIME, READ BACK  
: CONTENTS AND MAKE SURE IT IS 177777. CHECK FOR NO CONTROLLER  
: ERROR, NO DATA LATE, INPUT READY SET, OUTPUT READY RESET.  
: NOW READ ANOTHER WORD FROM THE SILO TO MAKE SURE DATA  
: LATE AND CONTROLLER ERROR SET.  
:\*\*\*\*\*  
TST65: SCOPE  
MOV #500, \$TIMES ;:DO 500. ITERATIONS  
MOV #CCLR, RKCS1(R2) ;:CLEAR RK611 CONTROLLER  
CLR RKWC(R2) ;:CLEAR WORD COUNT REG  
MOV #177777, CONFIG ;:LOAD CONFIGURATION FOR PRINT OUT  
MOV #177777, RKDB(R2) ;:WRITE DATA BUFFER WITH ALL 1'S  
MOV RKCS1(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG 1  
CMP #RDY, \$BDDAT ;:CHECK IF CS1 CORRECT  
BEQ 1\$ ;:YES, CHECK BUS AND REG  
MOV #RDY, \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1017, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV RKBA(R2), \$BDDAT ;:STORE BUS AND  
BEQ 2\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1019, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV RKWC(R2), \$BDDAT ;:STORE WORD COUNT REG  
BEQ 3\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1018, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV RKDA(R2), \$BDDAT ;:STORE DESK ADDRESS REG  
BEQ 4\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1020, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV RKASOF(R2), \$BDDAT ;:STORE ATTENTION SUMMARY AND OFFSET  
BEQ 5\$ ;:CHECK IF ZERO  
CLR \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1029, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV #5, R0 ;:LOAD COUNTER TO WAIT FOR OUTPUT READY  
DEC R0  
BNE 6\$  
MOV RKCS2(R2), \$BDDAT ;:STORE COMMAND AND STATUS REG. 2  
CMP #IR!OR, \$BDDAT ;:CHECK IF CS2 CORRECT  
BEQ 7\$ ;:YES, CONTINUE TEST  
MOV #IR!OR, \$GDDAT ;:LOAD EXPECTED CONTENTS  
MOV #EM1004, EM3N+2 ;:LOAD ERROR MESSAGE  
ERROR 3  
MOV RKDS(R2), \$BDDAT ;:STORE DRIVE STATUS REG.



7417	050056	001406				BEQ	8\$	:CHECK IF ZERO
7418	050060	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7419	050064	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
7420	050072	104003				ERROR	3	
7421	050074	016237	000014	001126	8\$:	MOV	RKER(R2),\$BDDAT	:STORE ERROR REGISTER
7422	050102	001406				BEQ	9\$	:CHECK IF ZERO
7423	050104	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7424	050110	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
7425	050116	104003				ERROR	3	
7426	050120	016237	000020	001126	9\$:	MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADDRESS REG.
7427	050126	001406				BEQ	11\$	:CHECK IF ZERO
7428	050130	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7429	050134	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
7430	050142	104003				ERROR	3	
7431	050144	016237	000026	001126	11\$:	MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG. 1
7432	050152	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
7433	050160	032737	020000	001126		BIT	#ECCW,\$BDDAT	
7434	050166	001403				BEQ	12\$	
7435	050170	052737	020000	001124		BIS	#ECCW,\$GDDAT	
7436	050176	023737	001124	001126	12\$:	CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
7437	050204	001404				BEQ	13\$	:YES, CONTINUE
7438	050206	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
7439	050214	104003				ERROR	3	
7440	050216	016237	000032	001126	13\$:	MOV	RKECPT(R2),\$BDDAT	:STORE ECC PATTERN REG.
7441	050224	001406				BEQ	14\$	:CHECK IF ZERO
7442	050226	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7443	050232	012737	067126	001322		MOV	#EM1008,EM3N+2	:LOAD ERROR MESSAGE
7444	050240	104003				ERROR	3	
7445	050242	016237	000030	001126	14\$:	MOV	RKECPS(R2),\$BDDAT	:STORE REC POSITION REC
7446	050250	022737	004066	001126		CMP	#4066,\$BDDAT	:CHECK IF POSITION CORRECT
7447	050256	001407				BEQ	15\$	:YES, CONTINUE
7448	050260	012737	004066	001124		MOV	#4066,\$GDDAT	:LOAD EXPECTED CONTENTS
7449	050266	012737	067764	001322		MOV	#EM1029,EM3N+2	:LOAD ERROR MESSAGE
7450	050274	104003				ERROR	3	
7451	050276	016237	000024	001126	15\$:	MOV	RKDB(R2),\$BDDAT	:STORE DATA BUFFER
7452	050304	022737	177777	001126		CMP	#177777,\$BDDAT	:CHECK CONTENTS CORRECT
7453	050312	001407				BEQ	16\$	:YES, CHECK TO MAKE SURE
7454								: DATA LATE RESET AND
7455								: CONTROLLER ERROR RESET
7456	050314	012737	177777	001124		MOV	#177777,\$GDDAT	:LOAD EXPECTED CONTENTS
7457	050322	012737	067126	001322		MOV	#EM1008,EM3N+2	:LOAD ERROR MESSAGE
7458	050330	104003				ERROR	3	
7459	050332	016237	000000	001700	16\$:	MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG. 1
7460	050340	016237	000010	001710		MOV	RKCS2(R2),T.CS2	:STORE COMMAND AND STATUS REG. 2
7461	050346	012737	000200	001740		MOV	#RDY,E.CS1	:LOAD EXPECTED CS1
7462	050354	012737	000100	001750		MOV	#IR,E.CS2	:LOAD EXPECTED CS2
7463	050362	023737	001740	001700		CMP	E.CS1,T.CS1	:CHECK IF CS1 CORRECT
7464	050370	001401				BEQ	17\$	:YES CHECK OUTPUT READY RESET
7465	050372	104022				ERROR	22	:ATTEMPTING TO READ SILO CONTAINING
7466								: ONE WORD - CS1 INCORRECT
7467	050374	023737	001750	001710	17\$:	CMP	E.CS2,T.CS2	:CHECK IN OUTPUT READY RESET
7468	050402	001401				BEQ	18\$	:YES, READ AN EXTRA WORD
7469	050404	104023				ERROR	23	:ATTEMPTING TO READ SILO COMMAND
7470								: ONE WORD - CS2 INCORRECT
7471	050406	005762	000024		18\$:	TST	RKDB(R2)	:READ DATA BUFFER
7472	050412	016237	000000	001700		MOV	RKCS1(R2),T.CS1	:STORE COMMAND AND STATUS REG. 7



```

7473 050420 016237 000010 001710      MOV      RKCS2(R2),T.CS2 ;STORE COMMAND AND STATUS REG. 2
7474 050426 012737 100200 001740      MOV      #CERR!RDY,E.CS1 ;LOAD EXPECTED CS1
7475 050434 012737 100100 001750      MOV      #DLT!IR,E.CS2  ;LOAD EXPECTED CS2
7476 050442 023737 001750 001710      CMP      E.CS2,T.CS2    ;CHECK IF DATA LATE SET
7477 050450 001401                                BEQ      19$            ;YES, CHECK IF CONTROLLER ERROR
7478                                ; SET
7479 050452 104016                                ERROR    16            ;ATTEMPTING TO READ SILO WHEN
7480                                ; EMPTY - CS2 INCORRECT
7481 050454 023737 001740 001700 19$:    CMP      E.CS1,T.CS1    ;CHECK IF CONTROLLER ERROR SET
7482 050462 001401                                BEQ      20$            ;YES, CLEAR CONTROLLER
7483 050464 104017                                ERROR    17            ;ATTEMPTING TO READ SILO WHEN
7484                                ; EMPTY - CS1 INCORRECT
7485 050466 012762 100000 000000 20$:    MOV      #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER
7486
7487                                ;*****
7488                                ;*TEST 66      ONE WORD SILO WRITE AND REG. INTERACTION (PART 1)
7489                                ;*
7490                                ;*      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7491                                ;*      ZERO.
7492                                ;*
7493                                ;*      WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
7494                                ;*      PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7495                                ;*      STATUS REGISTER 2. IF NOT WAIT.
7496                                ;*
7497                                ;*      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
7498                                ;*      IF CORRECT.
7499                                ;*
7500                                ;*      THE FOLLOWING CONFIGURATIONS ARE USED:
7501                                ;*
7502                                ;*      000000 000010 000200 004000 100000
7503                                ;*      000001 000020 000400 010000
7504                                ;*      000002 000040 001000 020000
7505                                ;*      000004 000100 002000 040000
7506                                ;*
7507                                ;*****
7508 050474 000004      TST66: SCOPE
7509 050476 012737 000764 001200      MOV      #500,$TIMES ;:DO 500. ITERATIONS
7510 050504 012737 065647 001320      MOV      #EM33,EM3N  ;:LOAD ERROR MESSAGE FOR PRINT OUT
7511 050512 012737 000001 002010      MOV      #000001,CONFIG ;:LOAD INITIAL CONFIGURATION
7512 050520 012701 000021                                MOV      #17.,R1      ;:LOAD CONFIGURATION COUNT
7513 050524 012737 050532 001110      MOV      #1$,$LPERR  ;:LOAD LOOP ON ERROR LOCATION FOR
7514                                ; SUBTEST LOOP
7515
7516 050532      1$:
7517 050532 012762 100000 000000      MOV      #CCLR,RKCS1(R2);CLEAR RK611 CONTROLLER
7518 050540 005062 000002                                CLR      RKWC(R2)     ;CLEAR WORD COUNT REG.
7519 050544 013762 002010 000024      MOV      CONFIG,RKDB(R2);WRITE DATA BUFFER
7520 050552 016237 000000 001126      MOV      RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
7521 050560 022737 000200 001126      CMP      #RDY,$BDDAT  ;CHECK IF CS1 CORRECT
7522 050566 001407                                BEQ      2$            ;YES, CHECK OTHER REGISTERS
7523 050570 012737 000200 001124      MOV      #RDY,$GDDAT  ;LOAD EXPECTED CONTENTS
7524 050576 012737 067376 001322      MOV      #EM1017,EM3N+2 ;LOAD ERROR MESSAGE
7525 050604 104003                                ERROR    3
7526 050606 016237 000004 001126 2$:    MOV      RKBA(R2),$BDDAT ;STORE BUS ADDRESS
7527 050614 001406                                BEQ      3$            ;CHECK IF ZERO
7528 050616 005037 001124                                CLR      $GDDAT      ;LOAD EXPECTED CONTENTS
  
```



7529	050622	012737	067447	001322		MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
7530	050630	104003				ERROR	3	
7531	050632	016237	000002	001126	3\$:	MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG.
7532	050640	001406				BEQ	4\$	:CHECK IF ZERO
7533	050642	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7534	050646	012737	067417	001322		MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
7535	050654	104003				ERROR	3	
7536	050656	016237	000006	001126	4\$:	MOV	RKDA(R2), \$BDDAT	:STORE DISK ADDRESS REG
7537	050664	001406				BEQ	5\$	:CHECK IF ZERO
7538	050666	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7539	050672	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
7540	050700	104003				ERROR	3	
7541	050702	016237	000016	001126	5\$:	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY AND OFFSET
7542	050710	001406				BEQ	6\$	:CHECK IF ZERO
7543	050712	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7544	050716	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
7545	050724	104003				ERROR	3	
7546	050726	012700	000005		6\$:	MOV	#5,RO	:LOAD COUNTER TO WAIT FOR
7547	050732	005300			7\$:	DEC	RO	: OUTPUT READY
7548	050734	001376				BNE	7\$	
7549	050736	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG 2
7550	050744	022737	000300	001126		CMP	#IR!OR, \$BDDAT	:CHECK IF CS2 CORRECT
7551	050752	001407				BEQ	8\$	:YES, CONTINUE TEST
7552	050754	012737	000300	001124		MOV	#IR!OR, \$GDDAT	:LOAD EXPECTED CONTENTS
7553	050762	012737	066766	001322		MOV	#EM1004,EM3N+2	:LOAD ERROR MESSAGE
7554	050770	104003				ERROR	3	
7555	050772	016237	000012	001126	8\$:	MOV	RKDS(R2), \$BDDAT	:STORE PRIVE STATUS REG
7556	051000	001406				BEQ	9\$	:CHECK IF ZERO
7557	051002	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7558	051006	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
7559	051014	104003				ERROR	3	
7560	051016	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REGISTER
7561	051024	001406				BEQ	10\$	:CHECK IF ZERO
7562	051026	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7563	051032	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
7564	051040	104003				ERROR	3	
7565	051042	016237	000020	001126	10\$:	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADDRESS REG
7566	051050	001406				BEQ	12\$	:CHECK IF ZERO
7567	051052	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7568	051056	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
7569	051064	104003				ERROR	3	
7570	051066	016237	000026	001126	12\$:	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG. 1
7571	051074	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
7572	051102	032737	020000	001126		BIT	#ECCW, \$BDDAT	
7573	051110	001403				BEQ	13\$	
7574	051112	052737	020000	001124		BIS	#ECCW, \$GDDAT	
7575	051120	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
7576	051126	001404				BEQ	14\$	:YES, CONTINUE
7577	051130	012737	067733	001322		MOV	#EM1026,EM3N+2	:LOAD ERROR MESSAGE
7578	051136	104003				ERROR	3	
7579	051140	016237	000032	001126	14\$:	MOV	RKECPT(R2), \$BDDAT	:STORE ECC PATTERN REG.
7580	051146	001406				BEQ	15\$	:CHECK IF ZERO
7581	051150	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7582	051154	012737	070011	001322		MOV	#EM1030,EM3N+2	:LOAD ERROR MESSAGE
7583	051162	104003				ERROR	3	
7584	051164	016237	000030	001126	15\$:	MOV	RKECPS(R2), \$BDDAT	:STORE ECC POSITION REG.



```

7585 051172 022737 004066 001126      CMP      #4066,$BDDAT      ;CHECK IF ECC POSITION REG CORRECT
7586 051200 001407                BEQ      16$              ;YES, CHECK DATA BUFFER
7587 051202 012737 004066 001124      MOV      #4066,$GDDAT      ;LOAD EXPECTED CONTENTS
7588 051210 012737 067764 001322      MOV      #EM1029,EM3N+2    ;LOAD ERROR MESSAGE
7589 051216 104003                ERROR    3
7590 051220 016237 000024 001126 16$:    MOV      RKDB(R2),$BDDAT    ;STORE DATA BUFFER
7591 051226 023737 002010 001126      CMP      CONFIG,$BDDAT    ;CHECK FOR CORRECT CONTENTS
7592 051234 001407                BEQ      17$              ;YES, CHECK IF FINISHED
7593 051236 013737 002010 001124      MOV      CONFIG,$GDDAT    ;LOAD EXPECTED CONTENTS
7594 051244 012737 067126 001322      MOV      #EM1008,EM3N+2    ;LOAD ERROR MESSAGE
7595 051252 104003                ERROR    3
7596 051254 104415                SCOPE1   ;CHECK IF LOOP ON ERROR
7597 051256 000241                CLC      ;SHIFT IN ZERO
7598 051260 006137 002010      ROL      CONFIG
7599 051264 005301                DEC      R1                ;CHECK IF FINISHED
7600 051266 001402                BEQ      TST67            ;:YES, GO ON TO NEXT TEST
7601 051270 000137 050532      JMP      1$
  
```

```

7602
7603      :*****
7604      *TEST 67      ONE WORD SILO WRITE AND REG. INTERACTION (PART 2)
7605      *
7606      *      ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7607      *      ZERO.
7608      *
7609      *      WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
7610      *      PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7611      *      STATUS REGISTER 2. IF NOT WAIT.
7612      *
7613      *      IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
7614      *      IF CORRECT.
7615      *
7616      *      THE FOLLOWING CONFIGURATIONS ARE USED:
7617      *
7618      *      177777 177767 177577 173777 077777
7619      *      177776 177757 177377 167777
7620      *      177775 177737 176777 157777
7621      *      177773 177677 175777 137777
7622      *
  
```

```

7623      :*****
7624      TST67:  SCOPE
7625      MOV      #500,$TIMES      ;:DO 500. ITERATIONS
7626      MOV      #EM33,EM3N      ;:LOAD ERROR MESSAGE FOR PRINT OUT
7627      MOV      #177776,CONFIG    ;:LOAD INITIAL CONFIGURATION
7628      MOV      #17,$R1          ;:LOAD CONFIGURATION COUNT
7629      MOV      #1$,$LPERR      ;:LOAD LOOP ON ERROR LOCATION FOR
7630      ;      SUBTEST LOOP
7631
7632      1$:
7633      MOV      #CCLR,RKCS1(R2)  ;:CLEAR RK611 CONTROLLER
7634      CLR      RKWC(R2)        ;:CLEAR WORD COUNT REG.
7635      MOV      CONFIG,RKDB(R2)  ;:WRITE DATA BUFFER
7636      MOV      RKCS1(R2),$BDDAT  ;:STORE COMMAND AND STATUS REG.1
7637      CMP      #RDY,$BDDAT      ;:CHECK IF CS1 CORRECT
7638      BEQ      2$              ;:YES, CHECK OTHER REGISTERS
7639      MOV      #RDY,$GDDAT      ;:LOAD EXPECTED CONTENTS
7640      MOV      #EM1017,EM3N+2    ;:LOAD ERROR MESSAGE
  
```



7641	051404	104003				ERROR	3	
7642	051406	016237	000004	001126	2\$:	MOV	RKBA(R2), \$BDDAT	:STORE BUS ADDRESS
7643	051414	001406				BEQ	3\$	:CHECK IF ZERO
7644	051416	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7645	051422	012737	067447	001322		MOV	#EM1019, EM3N+2	:LOAD ERROR MESSAGE
7646	051430	104003				ERROR	3	
7647	051432	016237	000002	001126	3\$:	MOV	RKWC(R2), \$BDDAT	:STORE WORD COUNT REG.
7648	051440	001406				BEQ	4\$	:CHECK IF ZERO
7649	051442	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7650	051446	012737	067417	001322		MOV	#EM1018, EM3N+2	:LOAD ERROR MESSAGE
7651	051454	104003				ERROR	3	
7652	051456	016237	000006	001126	4\$:	MOV	RKDA(R2), \$BDDAT	:STORE DISK ADDRESS REG
7653	051464	001406				BEQ	5\$	:CHECK IF ZERO
7654	051466	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7655	051472	012737	067474	001322		MOV	#EM1020, EM3N+2	:LOAD ERROR MESSAGE
7656	051500	104003				ERROR	3	
7657	051502	016237	000016	001126	5\$:	MOV	RKASOF(R2), \$BDDAT	:STORE ATTENTION SUMMARY AND OFFSET
7658	051510	001406				BEQ	6\$	:CHECK IF ZERO
7659	051512	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7660	051516	012737	067640	001322		MOV	#EM1024, EM3N+2	:LOAD ERROR MESSAGE
7661	051524	104003				ERROR	3	
7662	051526	012700	000005		6\$:	MOV	#5, R0	:LOAD COUNTER TO WAIT FOR
7663	051532	005300			7\$:	DEC	R0	: OUTPUT READY
7664	051534	001376				BNE	7\$	
7665	051536	016237	000010	001126		MOV	RKCS2(R2), \$BDDAT	:STORE COMMAND AND STATUS REG 2
7666	051544	022737	000300	001126		CMP	#IR!OR, \$BDDAT	:CHECK IF CS2 CORRECT
7667	051552	001407				BEQ	8\$	:YES, CONTINUE TEST
7668	051554	012737	000300	001124		MOV	#IR!OR, \$GDDAT	:LOAD EXPECTED CONTENTS
7669	051562	012737	066766	001322		MOV	#EM1004, EM3N+2	:LOAD ERROR MESSAGE
7670	051570	104003				ERROR	3	
7671	051572	016237	000012	001126	8\$:	MOV	RKDS(R2), \$BDDAT	:STORE PRIVE STATUS REG
7672	051600	001406				BEQ	9\$	:CHECK IF ZERO
7673	051602	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7674	051606	012737	067553	001322		MOV	#EM1022, EM3N+2	:LOAD ERROR MESSAGE
7675	051614	104003				ERROR	3	
7676	051616	016237	000014	001126	9\$:	MOV	RKER(R2), \$BDDAT	:STORE ERROR REGISTER .
7677	051624	001406				BEQ	10\$	:CHECK IF ZERO
7678	051626	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7679	051632	012737	067611	001322		MOV	#EM1023, EM3N+2	:LOAD ERROR MESSAGE
7680	051640	104003				ERROR	3	
7681	051642	016237	000020	001126	10\$:	MOV	RKDCYL(R2), \$BDDAT	:STORE CYLINDER ADDRESS REG
7682	051650	001406				BEQ	12\$-	:CHECK IF ZERO
7683	051652	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7684	051656	012737	067706	001322		MOV	#EM1025, EM3N+2	:LOAD ERROR MESSAGE
7685	051664	104003				ERROR	3	
7686	051666	016237	000026	001126	12\$:	MOV	RKMR1(R2), \$BDDAT	:STORE MAINTENANCE REG. 1
7687	051674	012737	002000	001124		MOV	#MEWD, \$GDDAT	:LOAD EXPECTED MR1
7688	051702	032737	020000	001126		BIT	#ECCW, \$BDDAT	
7689	051710	001403				BEQ	13\$	
7690	051712	052737	020000	001124		BIS	#ECCW, \$GDDAT	
7691	051720	023737	001124	001126	13\$:	CMP	\$GDDAT, \$BDDAT	:CHECK IF MR1 CORRECT
7692	051726	001404				BEQ	14\$	:YES, CONTINUE
7693	051730	012737	067733	001322		MOV	#EM1026, EM3N+2	:LOAD ERROR MESSAGE
7694	051736	104003				ERROR	3	
7695	051740	016237	000032	001126	14\$:	MOV	RKE(PTR), \$BDDAT	:STORE ECC PATTERN REG.
7696	051746	001406				BEQ	15\$	:CHECK IF ZERO



```

7697 051750 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
7698 051754 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
7699 051762 104003 ERROR 3
7700 051764 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
7701 051772 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION REG CORRECT
7702 052000 001407 BEQ 16$ ;YES, CHECK DATA BUFFER
7703 052002 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED CONTENTS
7704 052010 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
7705 052016 104003 ERROR 3
7706 052020 016237 000024 001126 16$: MOV RKDB(R2),$BDDAT ;STORE DATA BUFFER
7707 052026 023737 002010 001126 CMP CONFIG,$BDDAT ;CHECK FOR CORRECT CONTENTS
7708 052034 001407 BEQ 17$ ;YES, CHECK IF FINISHED
7709 052036 013737 002010 001124 MOV CONFIG,$GDDAT ;LOAD EXPECTED CONTENTS
7710 052044 012737 067126 001322 MOV #EM1008,EM3N+2 ;LOAD ERROR MESSAGE
7711 052052 104003 ERROR 3
7712 052054 104415 17$: SCOP1 ;CHECK IF LOOP ON ERROR
7713 052056 000261 SEC ;SHIFT IN ONE
7714 052060 006137 002010 ROL CONFIG
7715 052064 005301 DEC R1 ;CHECK IF FINISHED
7716 052066 001402 BEQ TST70 ;:YES, GO ON TO NEXT TEST
7717 052070 000137 051332 JMP 1$
  
```

```

7718
7719
7720 :*****
7721 :*TEST 70 ONE WORD SILO WRITE AND REG. INTERACTION (PART 3)
7722 :*
7723 :* ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7724 :* ZERO.
7725 :*
7726 :* WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
7727 :* PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7728 :* STATUS REGISTER 2. IF NOT WAIT.
7729 :*
7730 :* IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
7731 :* IF CORRECT.
7732 :*
7733 :* THE FOLLOWING CONFIGURATIONS ARE USED:
7734 :*
7735 :* 000001 000037 000777 017777 000000
7736 :* 000003 000077 001777 037777
7737 :* 000007 000177 003777 077777
7738 :* 000017 000377 007777 177777
7739 :*****
  
```

```

7740 052074 000004 TST70: SCOPE
7741 052076 012737 000764 001200 MOV #500,$TIMES ;:DO 500. ITERATIONS
7742 052104 012737 065647 001320 MOV #EM33,EM3N ;LOAD ERROR MESSAGE FOR PRINT OUT
7743 052112 005037 002010 CLR CONFIG ;LOAD INITIAL CONFIGURATION
7744 052116 012701 000021 MOV #17,R1 ;LOAD CONFIGURATION COUNT
7745 052122 012737 052130 001110 MOV #1,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
7746 ; SUBTEST LOOP
7747
7748 052130 1$: MOV #CLR,RKCS1(R2) ;CLEAR RK611 CONTROLLER
7749 052130 012762 100000 000000 CLR RKWC(R2) ;CLEAR WORD COUNT REG.
7750 052136 005062 000002 MOV CONFIG,RKDB(R2) ;WRITE DATA BUFFER
7751 052142 013762 002010 000024 MOV RKCS1(R2),$BDDAT ;STORE COMMAND AND STATUS REG.1
7752 052150 016237 000000
  
```



7753	052156	022737	000200	001126	CMP	#RDY,\$BDDAT	:CHECK IF CS1 CORRECT
7754	052164	001407			BEQ	2\$	:YES, CHECK OTHER REGISTERS
7755	052166	012737	000200	001124	MOV	#RDY,\$GDDAT	:LOAD EXPECTED CONTENTS
7756	052174	012737	067376	001322	MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
7757	052202	104003			ERROR	3	
7758	052204	016237	000004	001126	2\$: MOV	RKBA(R2),\$BDDAT	:STORE BUS ADDRESS
7759	052212	001406			BEQ	3\$	:CHECK IF ZERO
7760	052214	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7761	052220	012737	067447	001322	MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
7762	052226	104003			ERROR	3	
7763	052230	016237	000002	001126	3\$: MOV	RKWC(R2),\$BDDAT	:STORE WORD COUNT REG.
7764	052236	001406			BEQ	4\$	:CHECK IF ZERO
7765	052240	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7766	052244	012737	067417	001322	MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
7767	052252	104003			ERROR	3	
7768	052254	016237	000006	001126	4\$: MOV	RKDA(R2),\$BDDAT	:STORE DISK ADDRESS REG
7769	052262	001406			BEQ	5\$	:CHECK IF ZERO
7770	052264	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7771	052270	012737	067474	001322	MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
7772	052276	104003			ERROR	3	
7773	052300	016237	000016	001126	5\$: MOV	RKASOF(R2),\$BDDAT	:STORE ATTENTION SUMMARY AND OFFSET
7774	052306	001406			BEQ	6\$	:CHECK IF ZERO
7775	052310	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7776	052314	012737	067640	001322	MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
7777	052322	104003			ERROR	3	
7778	052324	012700	000005		6\$: MOV	#5,R0	:LOAD COUNTER TO WAIT FOR
7779	052330	005300			7\$: DEC	R0	: OUTPUT READY
7780	052332	001376			BNE	7\$	
7781	052334	016237	000010	001126	MOV	RKCS2(R2),\$BDDAT	:STORE COMMAND AND STATUS REG 2
7782	052342	022737	000300	001126	CMP	#IR!OR,\$BDDAT	:CHECK IF CS2 CORRECT
7783	052350	001407			BEQ	8\$	:YES, CONTINUE TEST
7784	052352	012737	000300	001124	MOV	#IR!OR,\$GDDAT	:LOAD EXPECTED CONTENTS
7785	052360	012737	066766	001322	MOV	#EM1004,EM3N+2	:LOAD ERROR MESSAGE
7786	052366	104003			ERROR	3	
7787	052370	016237	000012	001126	8\$: MOV	RKDS(R2),\$BDDAT	:STORE PRIVE STATUS REG
7788	052376	001406			BEQ	9\$	:CHECK IF ZERO
7789	052400	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7790	052404	012737	067553	001322	MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
7791	052412	104003			ERROR	3	
7792	052414	016237	000014	001126	9\$: MOV	RKER(R2),\$BDDAT	:STORE ERROR REGISTER
7793	052422	001406			BEQ	10\$	:CHECK IF ZERO
7794	052424	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7795	052430	012737	067611	001322	MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
7796	052436	104003			ERROR	3	
7797	052440	016237	000020	001126	10\$: MOV	RKDCYL(R2),\$BDDAT	:STORE CYLINDER ADDRESS REG
7798	052446	001406			BEQ	12\$	:CHECK IF ZERO
7799	052450	005037	001124		CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7800	052454	012737	067706	001322	MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
7801	052462	104003			ERROR	3	
7802	052464	016237	000026	001126	12\$: MOV	RKMR1(R2),\$BDDAT	:STORE MAINTENANCE REG. 1
7803	052472	012737	002000	001124	MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
7804	052500	032737	020000	001126	BIT	#ECCW,\$BDDAT	
7805	052506	001403			BEQ	13\$	
7806	052510	052737	020000	001124	BIS	#ECCW,\$GDDAT	
7807	052516	023737	001124	001126	13\$: CMP	\$GDDAT,\$BDDAT	:CHECK IF MR1 CORRECT
7808	052524	001404			BEQ	14\$	:YES,CONTINUE



```

7809 052526 012737 067733 001322      MOV      #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
7810 052534 104003                ERROR    3
7811 052536 016237 000032 001126 14$:  MOV      RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
7812 052544 001406                BFO     15$ ;CHECK IF ZERO
7813 052546 005037 001124                CLR      $GDDAT ;LOAD EXPECTED CONTENTS
7814 052552 012737 070011 001322      MOV      #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
7815 052560 104003                ERROR    3
7816 052562 016237 000030 001126 15$:  MOV      RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
7817 052570 022737 004066 001126      CMP      #4066,$BDDAT ;CHECK IF ECC POSITION REG CORRECT
7818 052576 001407                BEQ     16$ ;YES, CHECK DATA BUFFER
7819 052600 012737 004066 001124      MOV      #4066,$GDDAT ;LOAD EXPECTED CONTENTS
7820 052606 012737 067764 001322      MOV      #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
7821 052614 104003                ERROR    3
7822 052616 016237 000024 001126 16$:  MOV      RKDB(R2),$BDDAT ;STORE DATA BUFFER
7823 052624 023737 002010 001126      CMP      CONFIG,$BDDAT ;CHECK FOR CORRECT CONTENTS
7824 052632 001407                BEQ     17$ ;YES, CHECK IF FINISHED
7825 052634 013737 002010 001124      MOV      CONFIG,$GDDAT ;LOAD EXPECTED CONTENTS
7826 052642 012737 067126 001322      MOV      #EM1008,EM3N+2 ;LOAD ERROR MESSAGE
7827 052650 104003                ERROR    3
7828 052652 104415                SCOPE1  ;CHECK IF LOOP ON ERROR
7829 052654 000261                SEC     ;SHIFT IN ONE
7830 052656 006137 002010                ROL     CONFIG
7831 052662 005301                DEC     R1 ;CHECK IF FINISHED
7832 052664 001402                BEQ     TST71 ;:YES, GO ON TO NEXT TEST
7833 052666 000137 052130                JMP     1$

```

```

7834
7835
7836 *****
7837 *TEST 71 ONE WORD SILO WRITE AND REG. INTERACTION (PART 4)
7838 *
7839 * ISSUE CONTROLLER CLEAR AND WRITE WORD COUNT REGISTER TO
7840 * ZERO.
7841 *
7842 * WRITE ONE WORD IN SILO. CHECK ALL REGISTERS FOR INTERACTION
7843 * PROBLEMS. CHECK THAT OUTPUT READY IS SET IN COMMAND AND
7844 * STATUS REGISTER 2. IF NOT WAIT.
7845 *
7846 * IF OUTPUT READY READY SETS READ BACK CONTENTS, AND CHECK
7847 * IF CORRECT.
7848 *
7849 * THE FOLLOWING CONFIGURATIONS ARE USED:
7850 *
7851 * 100000 174000 177600 177770 000000
7852 * 140000 176000 177700 177774
7853 * 160000 177000 177740 177776
7854 * 170000 177400 177760 177777
7855 *

```

```

7856 052672 000004                TST71: SCOPE
7857 052674 012737 000764 001200      MOV      #500,$TIMES ;:DO 500. ITERATIONS
7858 052702 012737 065647 001320      MOV      #EM33,EM3N ;LOAD ERROR MESSAGE FOR PRINT OUT
7859 052710 005037 002010                CLR      CONFIG ;LOAD INITIAL CONFIGURATION
7860 052714 012701 000021                MOV      #17,R1 ;LOAD CONFIGURATION COUNT
7861 052720 012737 052726 001110      MOV      #1,$LPERR ;LOAD LOOP ON ERROR LOCATION FOR
7862 ; SUBTEST LOOP
7863
7864 052726                1$:

```



D 12

CZR6AD0 RK611 DSKLS CTRL PRT1 MACY11 30(1046) 14-SEP-81 15:04 PAGE 147  
 CZR6AD.P11 14-SEP-81 13:43 171 ONE WORD SILO WRITE AND REG. INTERACT.ON (PART 4) SEQ 0146

7865	052726	012762	100000	000000		MOV	#CCLR,RKCS1(R2)	:CLEAR RK611 CONTROLLER
7866	052734	005062	000002			CLR	RKWC(R2)	:CLEAR WORD REG.
7867	052740	013762	002010	000024		MOV	CONFIG,RKDB(R2)	:WRITE DATA
7868	052746	016237	000000	001126		MOV	RKCS1(R2),SBDDAT	:STORE AND STATUS REG.1
7869	052754	022737	000200	001126		CMP	#RDY,SBDDAT	:CHECK IF CS
7870	052762	001407				BEQ	2\$	:YES, CHECK DI. *STERS
7871	052764	012737	000200	001124		MOV	#RDY,\$GDDAT	:LOAD EXPECTED
7872	052772	012737	067376	001322		MOV	#EM1017,EM3N+2	:LOAD ERROR MESSAGE
7873	053000	104003				ERROR	3	
7874	053002	016237	000004	001126	2\$:	MOV	RKBA(R2),SBDDAT	:STORE BUS ADDRESS
7875	053010	001406				BEQ	3\$	:CHECK IF ZERO
7876	053012	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7877	053016	012737	067447	001322		MOV	#EM1019,EM3N+2	:LOAD ERROR MESSAGE
7878	053024	104003				ERROR	3	
7879	053026	016237	000002	001126	3\$:	MOV	RKWC(R2),SBDDAT	:STORE WORD COUNT REG.
7880	053034	001406				BEQ	4\$	:CHECK IF ZERO
7881	053036	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7882	053042	012737	067417	001322		MOV	#EM1018,EM3N+2	:LOAD ERROR MESSAGE
7883	053050	104003				ERROR	3	
7884	053052	016237	000006	001126	4\$:	MOV	RKDA(R2),SBDDAT	:STORE DISK ADDRESS REG
7885	053060	001406				BEQ	5\$	:CHECK IF ZERO
7886	053062	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7887	053066	012737	067474	001322		MOV	#EM1020,EM3N+2	:LOAD ERROR MESSAGE
7888	053074	104003				ERROR	3	
7889	053076	016237	000016	001126	5\$:	MOV	RKASOF(R2),SBDDAT	:STORE ATTENTION SUMMARY AND OFFSET
7890	053104	001406				BEQ	6\$	:CHECK IF ZERO
7891	053106	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7892	053112	012737	067640	001322		MOV	#EM1024,EM3N+2	:LOAD ERROR MESSAGE
7893	053120	104003				ERROR	3	
7894	053122	012700	000005		6\$:	MOV	#5,R0	:LOAD COUNTER TO WAIT FOR
7895	053126	005300			7\$:	DEC	R0	: OUTPUT READY
7896	053130	001376				BNE	7\$	
7897	053132	016237	000010	001126		MOV	RKCS2(R2),SBDDAT	:STORE COMMAND AND STATUS REG 2
7898	053140	022737	000300	001126		CMP	#IR!OR,SBDDAT	:CHECK IF CS2 CORRECT
7899	053146	001407				BEQ	8\$	:YES, CONTINUE TEST
7900	053150	012737	000300	001124		MOV	#IR!OR,\$GDDAT	:LOAD EXPECTED CONTENTS
7901	053156	012737	066766	001322		MOV	#EM1004,EM3N+2	:LOAD ERROR MESSAGE
7902	053164	104003				ERROR	3	
7903	053166	016237	000012	001126	8\$:	MOV	RKDS(R2),SBDDAT	:STORE PRIVE STATUS REG
7904	053174	001406				BEQ	9\$	:CHECK IF ZERO
7905	053176	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7906	053202	012737	067553	001322		MOV	#EM1022,EM3N+2	:LOAD ERROR MESSAGE
7907	053210	104003				ERROR	3	
7908	053212	016237	000014	001126	9\$:	MOV	RKER(R2),SBDDAT	:STORE ERROR REGISTER
7909	053220	001406				BEQ	10\$	:CHECK IF ZERO
7910	053222	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7911	053226	012737	067611	001322		MOV	#EM1023,EM3N+2	:LOAD ERROR MESSAGE
7912	053234	104003				ERROR	3	
7913	053236	016237	000020	001126	10\$:	MOV	RKDCYL(R2),SBDDAT	:STORE CYLINDER ADDRESS REG
7914	053244	001406				BEQ	12\$	:CHECK IF ZERO
7915	053246	005037	001124			CLR	\$GDDAT	:LOAD EXPECTED CONTENTS
7916	053252	012737	067706	001322		MOV	#EM1025,EM3N+2	:LOAD ERROR MESSAGE
7917	053260	104003				ERROR	3	
7918	053262	016237	000026	001126	12\$:	MOV	RKMR1(R2),SBDDAT	:STORE MAINTENANCE REG. 1
7919	053270	012737	002000	001124		MOV	#MEWD,\$GDDAT	:LOAD EXPECTED MR1
7920	053276	032737	020000	001126		BIT	#ECCW,SBDDAT	



```

7921 053304 001403 BEQ 13$
7922 053306 052737 020000 001124 BIS #ECCW,$GDDAT
7923 053314 023737 001124 001126 13$: CMP $GDDAT,$BDDAT ;CHECK IF MR1 CORRECT
7924 053322 001404 BEQ 14$ ;YES,CONTINUE
7925 053324 012737 067733 001322 MOV #EM1026,EM3N+2 ;LOAD ERROR MESSAGE
7926 053332 104003 ERROR 3
7927 053334 016237 000032 001126 14$: MOV RKECPT(R2),$BDDAT ;STORE ECC PATTERN REG.
7928 053342 001406 BEQ 15$ ;CHECK IF ZERO
7929 053344 005037 001124 CLR $GDDAT ;LOAD EXPECTED CONTENTS
7930 053350 012737 070011 001322 MOV #EM1030,EM3N+2 ;LOAD ERROR MESSAGE
7931 053356 104003 ERROR 3
7932 053360 016237 000030 001126 15$: MOV RKECPS(R2),$BDDAT ;STORE ECC POSITION REG.
7933 053366 022737 004066 001126 CMP #4066,$BDDAT ;CHECK IF ECC POSITION REG CORRECT
7934 053374 001407 BEQ 16$ ;YES, CHECK DATA BUFFER
7935 053376 012737 004066 001124 MOV #4066,$GDDAT ;LOAD EXPECTED CONTENTS
7936 053404 012737 067764 001322 MOV #EM1029,EM3N+2 ;LOAD ERROR MESSAGE
7937 053412 104003 ERROR 3
7938 053414 016237 000024 001126 16$: MOV RKDB(R2),$BDDAT ;STORE DATA BUFFER
7939 053422 023737 002010 001126 CMP CONFIG,$BDDAT ;CHECK FOR CORRECT CONTENTS
7940 053430 001407 BEQ 17$ ;YES, CHECK IF FINISHED
7941 053432 013737 002010 001124 MOV CONFIG,$GDDAT ;LOAD EXPECTED CONTENTS
7942 053440 012737 067126 001322 MOV #EM1008,EM3N+2 ;LOAD ERROR MESSAGE
7943 053446 104003 ERROR 3
7944 053450 104415 17$: SCOPI ;CHECK IF LOOP ON ERROR
7945 053452 000261 SEC ;SHIFT IN ONE
7946 053454 006037 002010 ROR CONFIG
7947 053460 005301 DEC R1 ;CHECK IF FINISHED
7948 053462 001402 BEQ TST72 ;:YES, GO ON TO NEXT TEST
7949 053464 000137 052726 JMP 1$
7950
7951 .....
7952 *TEST 72 SILO FILL
7953 *
7954 * THIS TEST WILL WRITE THE SILO WITH 66 DIFFERENT PATTERNS
7955 * CHECK INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
7956 * WORD WRITTEN. IT WILL THEN READ ALL 66 WORDS BACK CHECKING
7957 * CONTENTS, INPUT READY, OUTPUT READY, AND DATA LATE FOR EACH
7958 * WORD READ. AN EXTRA READ IS THEN DONE TO MAKE SURE THE
7959 * SILO IS EMPTY.
7960 *
7961 .....
7962 053470 000004 TST72: SCOPE
7963 053472 012737 000764 001200 MOV #500,$TIMES ;:DO 500. ITERATIONS
7964 053500 013702 001270 MOV $BASE,R2 ;:LOAD RK611 BASE
7965 053504 012762 100000 000000 MOV #CCLR,RKCS1(R2) ;:CLEAR RK611 CONTROLLER
7966 053512 012703 071036 MOV #SILO,R3 ;:LOAD ADDRESS OF INPUT DATA
7967 053516 005037 002022 CLR SILCNT ;:CLEAR SILO COUNT
7968 053522 012362 000024 MOV (R3)+,RKDB(R2) ;:LOAD SILO
7969 053526 012700 000020 MOV #20,R0 ;:WAIT FOR OUTPUT READY
7970 053532 005300 1$: DEC R0
7971 053534 001376 BNE 1$
7972 053536 016237 000000 001700 MOV RKCS1(R2),T.CS1 ;:STORE COMMAND AND STATUS REG 1
7973 053544 016237 000010 001710 MOV RKCS2(R2),T.CS2 ;:STORE COMMAND AND STATUS REG 2
7974 053552 012737 000200 001740 MOV #RDY,E.CS1 ;:LOAD EXPECTED COMMAND AND STATUS REG. 1
7975 053560 012737 000300 001750 MOV #IR!OR,E.CS2 ;:LOAD EXPECTED COMMAND AND STATUS REG 2
7976 053566 023737 001740 001700 CMP E.CS1,T.CS1 ;:CHECK IF CS1 CORRECT
  
```



7977	053574	001401				BEQ	2\$		:YES, CONTINUE
7978	053576	104024				ERROR	24		:ATTEMPTING TO WRITE SILO-CS1 INCORRECT
7979	053600	023737	001750	001710	2\$:	CMP	E.CS2,T.CS2		:CHECK IF CS2 CORRECT
7980	053606	001401				BEQ	3\$		:YES, CONTINUE
7981	053610	104025				ERROR	25		:ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7982	053612	005237	002022		3\$:	INC	SILCNT		:INCREMENT SILO COUNT
7983	053616	012362	000024			MOV	(R3)+,RKDB(R2)		:LOAD SILO
7984	053622	016237	000000	001700		MOV	RKCS1(R2),T.CS1		:STORE COMMAND AND STATUS REG. 1
7985	053630	016237	000010	001710		MOV	RKCS2(R2),T.CS2		:STORE COMMAND AND STATUS REG.2
7986	053636	023737	001740	001700		CMP	E.CS1,T.CS1		:CHECK IF CS1 CORRECT
7987	053644	001401				BEQ	4\$		:YES, CONTINUE
7988	053646	104024				ERROR	24		:ATTEMPTING TO WRITE SILO- CS1 INCORRECT
7989	053650	023737	001750	001710	4\$:	CMP	E.CS2,T.CS2		:CHECK IF CS2 CORRECT
7990	053656	001401				BEQ	5\$		:YES, CONTINUE
7991	053660	104025				ERROR	25		:ATTEMPTING TO WRITE SILO-CS2 INCORRECT
7992	053662	022737	000100	002022	5\$:	CMP	#64.,SILCNT		:CHECK IF ALL EXCEPT LAST WORD WRITTEN
7993	053670	001350				BNE	3\$		:YES, GO WRITE NEXT WORD
7994	053672	005237	002022			INC	SILCNT		:INCREMENT SILO COUNT
7995	053676	011362	000024			MOV	(R3),RKDB(R2)		:LOAD SILO
7996	053702	016237	000000	001700		MOV	RKCS1(R2),T.CS1		:STORE COMMAND AND STATUS REG. 1
7997	053710	016237	000010	001710		MOV	RKCS2(R2),T.CS2		:STORE COMMAND AND STATUS REG. 2
7998	053716	012737	000200	001750		MOV	#OR,E.CS2		:LOAD EXPECTED CS2
7999	053724	023737	001740	001700		CMP	E.CS1,T.CS1		:CHECK IF CS1 CORRECT
8000	053732	001401				BEQ	6\$		:YES, CONTINUE
8001	053734	104024				ERROR	24		:ATTEMPTING TO WRITE SILO-CS1 INCORRECT
8002	053736	023737	001750	001710	6\$:	CMP	E.CS2,T.CS2		:CHECK IF CS2 CORRECT
8003	053744	001401				BEQ	7\$		:YES, UNLOAD SILO
8004	053746	104025				ERROR	25		:ATTEMPTING TO WRITE SILO-CS1 INCORRECT
8005	053750	005037	002022		7\$:	CLR	SILCNT		:CLEAR SILO COUNT
8006	053754	012703	071036			MOV	#SILO,R3		:LOAD ADDRESS OF INPUT DATA
8007	053760	012737	070300	001750		MOV	#IR!OR,E.CS2		:LOAD EXPECTED CS2
8008	053766	016237	000024	001126		MOV	RKDB(R2),SBDDAT		:STORE NEXT WORD ON SILO
8009	053774	012700	000020			MOV	#20,RO		:WAIT FOR INPUT READY
8010	054000	005300			8\$:	DEC	RO		
8011	054002	001376				BNE	8\$		
8012	054004	000403				BR	10\$		:CONTINUE WITH TEST
8013									
8014	054006	016237	000024	001126	9\$:	MOV	RKDB(R2),SBDDAT		:STORE NEXT WORD ON SILO
8015	054014	016237	000000	001700	10\$:	MOV	RKCS1(R2),T.CS1		:STORE COMMAND AND STATUS REG 1
8016	054022	016237	000010	001710		MOV	RKCS2(R2),T.CS2		:STORE COMMAND AND STATUS REG 2
8017	054030	012337	001124			MOV	(R3)+,\$GDDAT		:LOAD EXPECTED DATA
8018	054034	023737	001740	001700		CMP	E.CS1,T.CS1		:CHECK IF CONTROLLER ERROR RESET
8019	054042	001401				BEQ	11\$		:YES, CONTINUE
8020	054044	104026				ERROR	26		:ATTEMPTING TO READ SILO-CS1 INCORRECT
8021	054046	023737	001750	001710	11\$:	CMP	E.CS2,T.CS2		:CHECK IF OUTPUT READY AND INPUT READY SET
8022	054054	001401				BEQ	12\$		:YES,CHECK DATA
8023	054056	104027				ERROR	27		:ATTEMPTING TO READ SILO-CS2 INCORRECT
8024	054060	023737	001124	001126	12\$:	CMP	\$GDDAT,\$BDDAT		:CHECK IF SILO CONTENTS CORRECT
8025	054066	001401				BEQ	13\$		:YES, CHECK IF LAST WORD
8026	054070	104030				ERROR	30		:ATTEMPTING TO READ SILO-RKDB INCORRECT
8027	054072	005237	002022		13\$:	INC	SILCNT		:SET UP FOR NEXT WORD
8028	054076	022737	000101	002022		CMP	#65.,SILCNT		:CHECK READY FOR LAST WORD
8029	054104	101340				BHI	9\$		:NO, READ NEXT WORD
8030	054106	103404				BLO	14\$		:CHECK IF SILO EMPTY
8031	054110	012737	000100	001750		MOV	#IR,E.CS2		:LOAD EXPECTED CS2
8032	054116	000733				BR	9\$		:READ LAST WORD







8089  
8090  
8091  
8092  
8093  
8094  
8095  
8096  
8097  
8098  
8099  
8100  
8101  
8102  
8103  
8104  
8105  
8106  
8107  
8108  
8109  
8110  
8111  
8112  
8113  
8114  
8115  
8116  
8117  
8118  
8119  
8120  
8121  
8122  
8123  
8124  
8125  
8126  
8127  
8128  
8129  
8130  
8131  
8132  
8133  
8134  
8135  
8136  
8137  
8138  
8139  
8140  
8141  
8142  
8143  
8144

```
*****  
: *TEST 74      INTERRUPT DUE TO DATA LATE  
: *  
: *      ALLOW RK611 INTERRUPTS.  SET INTERRUPT ENABLE.  
: *      NOW READ ONE WORD FROM DATA BUFFER AND MAKE SURE THAT  
: *      DATA LATE CAUSES INTERRUPT.  BEFORE CLEARING ERROR ALLOW  
: *      RK611 INTERRUPTS AND MAKE SURE IT DOES NOT OCCUR AGAIN.  
: *      NOW CLEAR CONTROLLER WITH A CONTROLLER CLEAR.  
: *  
*****  
TST74:  SCOPE  
MOV      #500, $TIMES      ;;DO 500. ITERATIONS  
MOV      $BASE, R2        ;:LOAD RK611 BASE  
MOV      #CCLR, RKCS1(R2) ;:CLEAR RK611  
MOV      RKVEC, R1        ;:LOAD VECTOR FOR EXPECTED INTERRUPT  
MOV      #10$, (R1)+  
MOV      #PR7, (R1)  
CLR      -(SP)            ;:LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV      #64$, -(SP)      ;:LOAD NEXT ADDRESS  
RTI  
  
64$:  
MOV      #IE, RKCS1(R2)   ;:SET INTERRUPT ENABLE  
TST      RKDB(R2)        ;:READ DATA BUFFER  
NOP  
MOV      #PR7, -(SP)     ;:ALLOW INTERRUPT TO OCCUR  
MOV      #1$, -(SP)     ;:LOCK OUT INTERRUPTS  
RTI  
  
1$:  
ERROR    33              ;:DATA LATE DID NOT CAUSE EXPECTED INTERRUPT  
BR       60$            ;:CLEAR UP FOR NEXT TEST  
  
10$:  
ADD      #4, SP          ;:ADJUST STACK  
MOV      #15$, @RKVEC    ;:LOAD VECTOR FOR EXPECTED INTERRUPT  
CLR      -(SP)          ;:LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV      #65$, -(SP)     ;:LOAD NEXT ADDRESS  
RTI  
;:CLEAR PSW  
  
65$:  
NOP  
MOV      #PR7, -(SP)     ;:ALLOW INTERRUPT TO OCCUR  
MOV      #60$, -(SP)     ;:LOCK OUT INTERRUPTS  
RTI  
;:CLEAN UP FOR NEXT TEST  
  
15$:  
ADD      #4, SP          ;:ADJUST STACK  
ERROR    34              ;:UNEXPECTED INTERRUPT DUE TO  
;: UNCLEARED CONTROLLER ERROR  
60$:  
MOV      #CCLR, RKCS1(R2) ;:CLEAR RK611 CONTROLLER  
MOV      RKVEC, R1      ;:RESTORE TRAP CATCHER  
MOV      R1, (R1)  
ADD      #2, (R1)+  
CLR      (R1)  
  
*****  
: *TEST 75      INTERRUPT CLEARING AND DATA LATE  
*****
```



8145  
8146  
8147  
8148  
8149  
8150  
8151  
8152  
8153 054572 000004  
8154 054574 012737 000144 001200  
8155 054602 013702 001270  
8156 054606 012762 100000 000000  
8157 054614 005762 000024  
8158 054620 012762 100000 000000  
8159 054626 013701 002000  
8160 054632 012721 054664  
8161 054636 012711 000340  
8162 054642 012762 000100 000000  
8163 054650 005046  
8164 054652 012746 054660  
8165 054656 000002  
8166  
8167 054660  
8168 054660 000240  
8169 054662 000403  
8170  
8171 054664 062706 000004  
8172 054670 104035  
8173  
8174  
8175 054672 013701 002000  
8176 054676 010111  
8177 054700 062721 000002  
8178 054704 005011  
8179  
8180  
8181  
8182  
8183  
8184  
8185  
8186  
8187  
8188  
8189  
8190 054706 000004  
8191 054710 012737 000764 001200  
8192 054716 013702 001270  
8193 054722 012762 100000 000000  
8194 054730 013701 002000  
8195 054734 012721 055016  
8196 054740 012711 000340  
8197 054744 005046  
8198 054746 012746 054754  
8199 054752 000002  
8200

\*\*\*\*\*  
: \* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. CREATE  
: \* A CONTROLLER ERROR (DATA LATE) BY READING THE DATA BUFFER  
: \* WHEN EMPTY. CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR.  
: \* SET INTERRUPT ENABLE AND LOWER PROCESSOR PRIORITY.  
: \* MAKE SURE AN INTERRUPT DOES NOT OCCUR.  
\*\*\*\*\*

TST75: SCOPE  
MOV #100, \$TIMES ;; DO 100. ITERATIONS  
MOV \$BASE, R2 ;: LOAD RK611 BASE  
MOV #CCLR, RKCS1(R2) ;: CLEAR RK611 CONTROLLER  
TST RKDB(R2) ;: CREATE DATA LATE  
MOV #CCLR, RKCS1(R2) ;: CLEAR DATA LATE  
MOV RKVEC, R1 ;: LOAD VECTOR FOR UNEXPECTED INTERRUPT  
MOV #5\$, (R1)+  
MOV #PR7, (R1)  
MOV #IE, RKCS1(R2) ;: SET INTERRUPT ENABLE  
CLR -(SP) ;: LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV #64\$, -(SP) ;: LOAD NEXT ADDRESS  
RTI ;: CLEAR PSW  
  
64\$:  
NOP ;: ALLOW INTERRUPT TO OCCUR  
BR 60\$ ;: CLEAN UP FOR NEXT TEST  
  
5\$:  
ADD #4, SP ;: ADJUST STACK  
ERROR 35 ;: CONTROLLER CLEAR DID NOT CLEAR  
;: PENDING INTERRUPT DUE  
;: TO CONTROLLER ERROR  
  
60\$:  
MOV RKVEC, R1 ;: RESTORE TRAP CATCHER  
MOV R1, (R1)  
ADD #2, (R1)+  
CLR (R1)

\*\*\*\*\*  
: \* TEST 76 INTERRUPT ENABLE AND DATA LATE  
: \*  
: \* CLEAR RK611 CONTROLLER WITH A CONTROLLER CLEAR. ALLOW  
: \* RK611 INTERRUPTS. READ DATA BUFFER TO GENERATE INTERRUPT  
: \* PENDING. MAKE SURE INTERRUPT DOES NOT OCCUR.  
: \*  
: \* NOW SET INTERRUPT ENABLE AND MAKE SURE INTERRUPTS OCCURS.  
\*\*\*\*\*

TST76: SCOPE  
MOV #500, \$TIMES ;; DO 500. ITERATIONS  
MOV \$BASE, R2 ;: LOAD RK611 BASE  
MOV #CCLR, RKCS1(R2) ;: CLEAR RK611 CONTROLLER  
MOV RKVEC, R1 ;: LOAD VECTOR FOR UNEXPECTED INTERRUPT  
MOV #10\$, (R1)+  
MOV #PR7, (R1)  
CLR -(SP) ;: LOAD STACK TO ALLOW ALL INTERRUPTS  
MOV #64\$, -(SP) ;: LOAD NEXT ADDRESS  
RTI ;: CLEAR PSW



```

8201 054754
8202 054754 005762 000024
8203 054760 000240
3204 054762 012777 055026 125010
8205 054770 012762 000100 000000
8206 054776 000240
8207 055000 012746 000340
8208 055004 012746 055012
8209 055010 000002
8210
8211
8212 055012 104037
8213
8214 055014 000406
8215
8216 055016 062706 000004
8217 055022 104036
8218
8219
8220 055024 000402
8221
8222 055026 062706 000004
8223 055032 012762 100000 000000
8224 055040 013701 002000
8225 055044 010111
8226 055046 062721 000002
8227 055052 005011

64$:
TST RKDB(R2) ;READ DATA BUFFER (GENERATE DATA LATE)
NOP ;ALLOW INTERRUPT TO OCCUR
MOV #15$,@RKVEC ;LOAD VECTOR FOR EXPECTED INTERRUPT
MOV #IE,RKCS1(R2) ;SET INTERRUPT ENABLE
NOP ;ALLOW INTERRUPT TO OCCUR
MOV #PR7,-(SP) ;LOCK OUT INTERRUPTS
MOV #1$,-(SP)
RTI

1$:
ERROR 37 ;INTERRUPT DID NOT OCCUR WHEN
; INTERRUPT ENABLE SET
BR 20$ ;CLEAN UP FOR NEXT TEST

10$:
ADD #4,SP ;ADJUST STACK
ERROR 36 ;CONTROLLER ERROR CAUSED INTERRUPT
; WITH INTERRUPT ENABLE RESET

BR 20$ ;CLEAN UP FOR NEXT TEST

15$:
ADD #4,SP ;ADJUST STACK
20$:
MOV #CCLR,RKCS1(R2) ;CLEAR RK611
MOV RKVEC,R1 ;RESTORE TRAP CATCHER
MOV R1,(R1)
ADD #2,(R1)+
CLR (R1)
  
```



8228  
8229  
8230  
8231  
8232  
8233  
8234  
8235  
8236  
8237 055054  
8238 055054 000004  
8239 055056 005037 001102  
8240 055062 005037 001200  
8241 055066 005237 001222  
8242 055072 042737 100000 001222  
8243 055100 005327  
8244 055102 000001  
8245 055104 003063  
8246 055106 012737  
8247 055110 000001  
8248 055112 055102  
8249 055114 104401 055122  
8250 055120 000407  
8251  
8252 055140  
8253 055140 013746 001222  
8254  
8255 055144 104405  
8256 055146 104401 055154  
8257 055152 000421  
8258  
8259 055216  
8260 055216 013746 001112  
8261  
8262 055222 104405  
8263 055224 104401 001211  
8264 055230 005037 001112  
8265 055234 013700 000042  
8266 055240 001405  
8267 055242 000005  
8268 055244 004710  
8269 055246 000240  
8270 055250 000240  
8271 055252 000240  
8272 055254  
8273 055254 000137  
8274 055256 003040  
8275 055260 377 377 000  
8276 055264  
8277  
8278  
8279  
8280 055264 012737 055336 000004  
8281 055272 012737 000340 000006  
8282 055300 012703 172100  
8283

```
.SBTTL END OF PASS ROUTINE

*****
*INCREMENT THE PASS NUMBER ($PASS)
*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYYY'
*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
*IF THERES A MONITOR GO TO IT
*IF THERE ISN'T JUMP TO NEWPAS

$EOP:
SCOPE
CLR $TSTNM ;;ZERO THE TEST NUMBER
CLR $TIMES ;;ZERO THE NUMBER OF ITERATIONS
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?

$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,@(PC)+ ;;RESTORE COUNTER

$ENDCT: .WORD 1
$EOPCT
TYPE ,65$ ;;TYPE ASCIZ STRING
BR 64$ ;;GET OVER THE ASCIZ
;;65$: .ASCIZ <12><15>/END PASS #/
64$:
MOV $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
;;TYPE PASS NUMBER
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE ,67$ ;;TYPE ASCIZ STRING
BR 66$ ;;GET OVER THE ASCIZ
;;67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
MOV $ERTTL,-(SP) ;;SAVE $ERTTL FOR TYPEOUT
;;TOTAL NUMBER OF ERRORS
TYPDS ;;GO TYPE--DECIMAL ASCII WITH SIGN
TYPE , $CRLF ;;TYPE CARRIAGE RETURN, LINE FEED
CLR $ERTTL ;;CLEAR ERROR TOTAL
$GET42: MOV @#42,R0 ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11

$DOAGN:
JMP @(PC)+ ;;RETURN
$RTNAD: .WORD NEWPAS
$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
.EVEN

.SBTTL CHECK FOR MEMORY CHECK ENABLE

CHKPAR: MOV #20$,ERRVEC ;;SET VECTOR ;FOR MEMORY PARTITY CHECK
MOV #PR7,ERRVEC+2
MOV #MEMBAS,R3 ;;LOAD REGISTER TO DETERMINE IF
;; MEMORY CHECK ENABLE AVAILIZGLE
```



```
8284 055304 012704 000020      MOV      #16,R4      ;LOAD COUNT
8285 055310 012723 000001      16$: MOV      #PAR.EN,(R3)+ ;ENABLE MEMORY CHECK
8286 055314 013737 055354 000114      MOV      MEMERR,MEMVEC ;LOAD MEMORY CHECK VECTOR
8287 055322 012737 000340 000116      MOV      #PR7,MEMVEC+2
8288 055330 005304      DEC      R4      ;CHECK IF FINISHED
8289 055332 001366      BNE     16$      ;NO, SET UP NEXT MEMORY PARITY MODULE
8290 055334 000401      BR      22$      ;RESTORE TRAP VECTOR
8291
8292 055336 022626      20$: CMP      (SP)+,(SP)+ ;ADJUST STACK
8293 055340 012737 000006 000004      22$: MOV      #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
8294 055346 005037 000006      CLR     ERRVEC+2
8295 055352 000207      RTS     PC      ;RETURN
8296
8297      .SBTTL MEMORY CHECK ENABLE TRAP
8298
8299 055354 012737 055370 001202 MEMERR: MOV      #10$, $ESCAPE ;LOAD ESCAPE
8300 055362 011637 002024      MOV      (SP),TRAPPC ;STORE PC
8301 055366 104040      ERROR   40      ;REPORT MEM PARITY ERROR
8302 055370 005037 001202      10$: CLR     $ESCAPE ;CLEAR ESCAPE
8303 055374 032777 001000 123536      BIT     #SW9,@SWR ;CHECK IF LOOP ON ERROR
8304 055402 001001      BNE     15$      ;YES, FORCE STACK AND TRY AGAIN
8305 055404 000002      RTI     ;NO, RETURN
8306
8307 055406 012706 001100      15$: MOV      #STACK,SP ;INITIALIZE STACK
8308 055412 000177 123472      JMP     @ $LPERR ;LOOP ON ERROR
8309
8310      .SBTTL SCOPE HANDLER ROUTINE
8311
8312      ;*****
8313      ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
8314      ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
8315      ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
8316      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
8317      ;*SW14=1      LOOP ON TEST
8318      ;*SW11=1      INHIBIT ITERATIONS
8319      ;*SW09=1      LOOP ON ERROR
8320      ;*SW08=1      LOOP ON TEST IN SWR<7:0>
8321      ;*CALL
8322      ;*      SCOPE      ;;SCOPE=IOT
8323
8324 055416      $SCOPE:
8325 055416 104407      CKSWR
8326 055420 032777 040000 123512 1$: BIT     #BIT14,@SWR ;:TEST FOR CHANGE IN SOFT-SWR
8327 055426 001131      BNE     $OVER ;:LOOP ON PRESENT TEST?
8328      ;#####START OF CODE FOR THE XOR TESTER#####
8329 055430 000416      $XTSTR: BR     6$ ;:IF RUNNING ON THE "XOR" TESTER CHANGE
8330      ;:THIS INSTRUCTION TO A "NOP" (NOP=240)
8331 055432 013746 000004      MOV     @#ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
8332 055436 012737 055456 000004      MOV     #5$,@#ERRVEC ;:SET FOR TIMEOUT
8333 055444 005737 177060      TST    @#177060 ;:TIME OUT ON XOR?
8334 055450 012637 000004      MOV     (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
8335 055454 000500      BR     $$VLAD ;:GO TO THE NEXT TEST
8336 055456 022626      5$: CMP     (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
8337 055460 012637 000004      MOV     (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
8338 055464 000440      BR     7$ ;:LOOP ON THE PRESENT TEST
8339 055466      6$:;#####END OF CODE FOR THE XOR TESTER#####
```



8340	055466	032777	000400	123444	BIT	#BIT08,@SWR	::LOOP ON SPEC. TEST?	
8341	055474	001421			BEQ	2\$	::BR IF NO	
8342	055476	005046			CLR	-(SP)	::CLEAR A TEMP. LOCATION	
8343	055500	117716	123434		MOVB	@SWR,(SP)	::PICKUP THE DESIRED TEST NUMBER	
8344	055504	001414			BEQ	8\$	::BRANCH IF BAD TEST NUMBER IN SWR	
8345	055506	022716	000076		CMP	#76,(SP)	::CHECK THE NUMBER IN THE SWR	
8346	055512	002411			BLT	8\$	::BRANCH IF TEST NUMBER IS OUT OF RANGE	
8347	055514	011637	001102		MOV	(SP),\$TSTNM	::UPDATE THE TEST NUMBER	
8348	055520	005316			DEC	(SP)	::BACKUP BY ONE	
8349	055522	006316			ASL	(SP)	::SCALE THE TEST NUMBER AS AN INDEX	
8350	055524	062716	055730		ADD	\$\$SW08TBL,(SP)	::FORM THE ADDRESS OF TEST POINTER	
8351	055530	013637	001106		MOV	@(SP)+,\$LPADR	::SET LOOP ADDRESS TO DESIRED TEST	
8352	055534	000466			BR	\$OVER	::GO LOOP ON THE TEST	
8353	055536	005726		8\$:	TST	(SP)+	::CLEAN THE BAD TEST NUMBER OFF OF THE STACK	
8354	055540	105737	001103	2\$:	TSTB	\$ERFLG	::HAS AN ERROR OCCURRED?	
8355	055544	001421			BEQ	3\$	::BR IF NO	
8356	055546	123737	001115	001103	CMPB	\$ERMAX,\$ERFLG	::MAX. ERRORS FOR THIS TEST OCCURRED?	
8357	055554	101015			BHI	3\$	::BR IF NO	
8358	055556	032777	001000	123354	BIT	#BIT09,@SWR	::LOOP ON ERROR?	
8359	055564	001404			BEQ	4\$	::BR IF NO	
8360	055566	013737	001110	001106	7\$:	MOV	\$LPERR,\$LPADR	::SET LOOP ADDRESS TO LAST SCOPE
8361	055574	000446			BR	\$OVER		
8362	055576	105037	001103	4\$:	CLRB	\$ERFLG	::ZERO THE ERROR FLAG	
8363	055602	005037	001200		CLR	\$TIMES	::CLEAR THE NUMBER OF ITERATIONS TO MAKE	
8364	055606	000415			BR	1\$	::ESCAPE TO THE NEXT TEST	
8365	055610	032777	004000	123322	3\$:	BIT	#BIT11,@SWR	::INHIBIT ITERATIONS?
8366	055616	001011			BNE	1\$	::BR IF YES	
8367	055620	005737	001222		TST	\$PASS	::IF FIRST PASS OF PROGRAM	
8368	055624	001406			BEQ	1\$	::INHIBIT ITERATIONS	
8369	055626	005237	001104		INC	\$ICNT	::INCREMENT ITERATION COUNT	
8370	055632	023737	001200	001104	CMP	\$TIMES,\$ICNT	::CHECK THE NUMBER OF ITERATIONS MADE	
8371	055640	002024			BGE	\$OVER	::BR IF MORE ITERATION REQUIRED	
8372	055642	012737	000001	001104	1\$:	MOV	#1,\$ICNT	::REINITIALIZE THE ITERATION COUNTER
8373	055650	013737	055726	001200	MOV	\$MXCNT,\$TIMES	::SET NUMBER OF ITERATIONS TO DO	
8374	055656	105237	001102		\$SVLAD: INCB	\$TSTNM	::COUNT TEST NUMBERS	
8375	055662	113737	001102	001220	MOVB	\$TSTNM,\$TESTN	::SET TEST NUMBER IN APT MAILBOX	
8376	055670	011637	001106		MOV	(SP),\$LPADR	::SAVE SCOPE LOOP ADDRESS	
8377	055674	011637	001110		MOV	(SP),\$LPERR	::SAVE ERROR LOOP ADDRESS	
8378	055700	005037	001202		CLR	\$ESCAPE	::CLEAR THE ESCAPE FROM ERROR ADDRESS	
8379	055704	112737	000001	001115	MOVB	#1,\$ERMAX	::ONLY ALLOW ONE(1) ERROR ON NEXT TEST	
8380	055712	013777	001102	123222	\$OVER: MOV	\$TSTNM,@DISPLAY	::DISPLAY TEST NUMBER	
8381	055720	013716	001106		MOV	\$LPADR,(SP)	::FUDGE RETURN ADDRESS	
8382	055724	000002			RTI		::FIXES PS	
8383	055726	003720			\$MXCNT: 2000.		::MAX. NUMBER OF ITERATIONS	
8384	055730				\$\$SW08TBL:			
8385	055730	003064			.WORD	TST1+2	::STARTING ADDRESS OF TEST 1	
8386	055732	003156			.WORD	TST2+2	::STARTING ADDRESS OF TEST 2	
8387	055734	003716			.WORD	TST3+2	::STARTING ADDRESS OF TEST 3	
8388	055736	004456			.WORD	TST4+2	::STARTING ADDRESS OF TEST 4	
8389	055740	005212			.WORD	TST5+2	::STARTING ADDRESS OF TEST 5	
8390	055742	005310			.WORD	TST6+2	::STARTING ADDRESS OF TEST 6	
8391	055744	006120			.WORD	TST7+2	::STARTING ADDRESS OF TEST 7	
8392	055746	006704			.WORD	TST10+2	::STARTING ADDRESS OF TEST 10	
8393	055750	007532			.WORD	TST11+2	::STARTING ADDRESS OF TEST 11	
8394	055752	010360			.WORD	TST12+2	::STARTING ADDRESS OF TEST 12	
8395	055754	011204			.WORD	TST13+2	::STARTING ADDRESS OF TEST 13	



8396	055756	012030	.WORD	TST14+2	:: STARTING ADDRESS OF TEST 14
8397	055760	012716	.WORD	TST15+2	:: STARTING ADDRESS OF TEST 15
8398	055762	013606	.WORD	TST16+2	:: STARTING ADDRESS OF TEST 16
8399	055764	014472	.WORD	TST17+2	:: STARTING ADDRESS OF TEST 17
8400	055766	015356	.WORD	TST20+2	:: STARTING ADDRESS OF TEST 20
8401	055770	016064	.WORD	TST21+2	:: STARTING ADDRESS OF TEST 21
8402	055772	016702	.WORD	TST22+2	:: STARTING ADDRESS OF TEST 22
8403	055774	017520	.WORD	TST23+2	:: STARTING ADDRESS OF TEST 23
8404	055776	020334	.WORD	TST24+2	:: STARTING ADDRESS OF TEST 24
8405	056000	021150	.WORD	TST25+2	:: STARTING ADDRESS OF TEST 25
8406	056002	021776	.WORD	TST26+2	:: STARTING ADDRESS OF TEST 26
8407	056004	022624	.WORD	TST27+2	:: STARTING ADDRESS OF TEST 27
8408	056006	023450	.WORD	TST30+2	:: STARTING ADDRESS OF TEST 30
8409	056010	024274	.WORD	TST31+2	:: STARTING ADDRESS OF TEST 31
8410	056012	025122	.WORD	TST32+2	:: STARTING ADDRESS OF TEST 32
8411	056014	025750	.WORD	TST33+2	:: STARTING ADDRESS OF TEST 33
8412	056016	026574	.WORD	TST34+2	:: STARTING ADDRESS OF TEST 34
8413	056020	027420	.WORD	TST35+2	:: STARTING ADDRESS OF TEST 35
8414	056022	030270	.WORD	TST36+2	:: STARTING ADDRESS OF TEST 36
8415	056024	031140	.WORD	TST37+2	:: STARTING ADDRESS OF TEST 37
8416	056026	032004	.WORD	TST40+2	:: STARTING ADDRESS OF TEST 40
8417	056030	032650	.WORD	TST41+2	:: STARTING ADDRESS OF TEST 41
8418	056032	033414	.WORD	TST42+2	:: STARTING ADDRESS OF TEST 42
8419	056034	034122	.WORD	TST43+2	:: STARTING ADDRESS OF TEST 43
8420	056036	034630	.WORD	TST44+2	:: STARTING ADDRESS OF TEST 44
8421	056040	035336	.WORD	TST45+2	:: STARTING ADDRESS OF TEST 45
8422	056042	036044	.WORD	TST46+2	:: STARTING ADDRESS OF TEST 46
8423	056044	036672	.WORD	TST47+2	:: STARTING ADDRESS OF TEST 47
8424	056046	037520	.WORD	TST50+2	:: STARTING ADDRESS OF TEST 50
8425	056050	040344	.WORD	TST51+2	:: STARTING ADDRESS OF TEST 51
8426	056052	041170	.WORD	TST52+2	:: STARTING ADDRESS OF TEST 52
8427	056054	042034	.WORD	TST53+2	:: STARTING ADDRESS OF TEST 53
8428	056056	042700	.WORD	TST54+2	:: STARTING ADDRESS OF TEST 54
8429	056060	043542	.WORD	TST55+2	:: STARTING ADDRESS OF TEST 55
8430	056062	044404	.WORD	TST56+2	:: STARTING ADDRESS OF TEST 56
8431	056064	045112	.WORD	TST57+2	:: STARTING ADDRESS OF TEST 57
8432	056066	045620	.WORD	TST60+2	:: STARTING ADDRESS OF TEST 60
8433	056070	046674	.WORD	TST61+2	:: STARTING ADDRESS OF TEST 61
8434	056072	047160	.WORD	TST62+2	:: STARTING ADDRESS OF TEST 62
8435	056074	047266	.WORD	TST63+2	:: STARTING ADDRESS OF TEST 63
8436	056076	047410	.WORD	TST64+2	:: STARTING ADDRESS OF TEST 64
8437	056100	047574	.WORD	TST65+2	:: STARTING ADDRESS OF TEST 65
8438	056102	050476	.WORD	TST66+2	:: STARTING ADDRESS OF TEST 66
8439	056104	051276	.WORD	TST67+2	:: STARTING ADDRESS OF TEST 67
8440	056106	052076	.WORD	TST70+2	:: STARTING ADDRESS OF TEST 70
8441	056110	052674	.WORD	TST71+2	:: STARTING ADDRESS OF TEST 71
8442	056112	053472	.WORD	TST72+2	:: STARTING ADDRESS OF TEST 72
8443	056114	054210	.WORD	TST73+2	:: STARTING ADDRESS OF TEST 73
8444	056116	054406	.WORD	TST74+2	:: STARTING ADDRESS OF TEST 74
8445	056120	054574	.WORD	TST75+2	:: STARTING ADDRESS OF TEST 75
8446	056122	054710	.WORD	TST76+2	:: STARTING ADDRESS OF TEST 76

::\*\*\*\*\*

8447  
8448  
8449 .SBTTL LOOP ON INTERNAL ERROR  
8450  
8451 056124 032777 001000 123006 SCOP18: BIT #SW9,@SWR :CHECK IF LOOP IN ERROR



```
8452 056132 001405 BEQ 5$ ;NO, CONTINUE TEST
8453 056134 105737 001103 TSTB $ERFLG ;CHECK IF ERROR OCCURRED
8454 056140 001402 BEQ 5$ ;NO, CONTINUE
8455 056142 013716 001110 MOV $LPERR,(SP) ;LOAD ERROR RETURN
8456 056146 000002 5$: RTI ;RETURN
8457 .SBTTL APT COMMUNICATIONS ROUTINE
8458
8459
8460 056150 112737 000001 056414 $ATY1: MOVB #1,$FFLG ;:TO REPORT FATAL ERROR
8461 056156 112737 000001 056412 $ATY3: MOVB #1,$MFLG ;:TO TYPE A MESSAGE
8462 056164 000403 BR $ATYC
8463 056166 112737 000001 056414 $ATY4: MOVB #1,$FFLG ;:TO ONLY REPORT FATAL ERROR
8464 056174 $ATYC:
8465 056174 010046 MOV R0,-(SP) ;:PUSH R0 ON STACK
8466 056176 010146 MOV R1,-(SP) ;:PUSH R1 ON STACK
8467 056200 105737 056412 TSTB $MFLG ;:SHOULD TYPE A MESSAGE?
8468 056204 001450 BEQ 5$ ;:IF NOT: BR
8469 056206 122737 000001 001234 CMPB #APTENV,$ENV ;:OPERATING UNDER APT?
8470 056214 001031 BNE 3$ ;:IF NOT: BR
8471 056216 132737 000100 001235 BITB #APTPOOL,$ENVM ;:SHOULD SPOOL MESSAGES?
8472 056224 001425 BEQ 3$ ;:IF NOT: BR
8473 056226 017600 000004 MOV @4(SP),R0 ;:GET MESSAGE ADDR.
8474 056232 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDR.
8475 056240 005737 001214 1$: TST $MSGTYPE ;:SEE IF DONE W/ LAST XMISSION?
8476 056244 001375 BNE 1$ ;:IF NOT: WAIT
8477 056246 010037 001230 MOV R0,$MSGAD ;:PUT ADDR IN MAILBOX
8478 056252 105720 2$: TSTB (R0)+ ;:FIND END OF MESSAGE
8479 056254 001376 BNE 2$
8480 056256 163700 001230 SUB $MSGAD,R0 ;:SUB START OF MESSAGE
8481 056262 006200 ASR R0 ;:GET MESSAGE LNGTH IN WORDS
8482 056264 010037 001232 MOV R0,$MSGGLT ;:PUT LENGTH IN MAILBOX
8483 056270 012737 000004 001214 MOV #4,$MSGTYPE ;:TELL APT TO TAKE MSG.
8484 056276 000413 BR 5$
8485 056300 017637 000004 056324 3$: MOV @4(SP),4$ ;:PUT MSG ADDR IN JSR LINKAGE
8486 056306 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDRESS
8487 056314 013746 177776 MOV 177776,-(SP) ;:PUSH 177776 ON STACK
8488 056320 004737 057100 JSR PC,$TYPE ;:CALL TYPE MACRO
8489 056324 000000 4$: .WORD 0
8490 056326 5$:
8491 056326 105737 056414 10$: TSTB $FFLG ;:SHOULD REPORT FATAL ERROR?
8492 056332 001416 BEQ 12$ ;:IF NOT: BR
8493 056334 005737 001234 TST $ENV ;:RUNNING UNDER APT?
8494 056340 001413 BEQ 12$ ;:IF NOT: BR
8495 056342 005737 001214 11$: TST $MSGTYPE ;:FINISHED LAST MESSAGE?
8496 056346 001375 BNE 11$ ;:IF NOT: WAIT
8497 056350 017637 000004 001216 MOV @4(SP),$FATAL ;:GET ERROR #
8498 056356 062766 000002 000004 ADD #2,4(SP) ;:BUMP RETURN ADDR.
8499 056364 005237 001214 INC $MSGTYPE ;:TELL APT TO TAKE ERROR
8500 056370 105037 056414 12$: CLRB $FFLG ;:CLEAR FATAL FLAG
8501 056374 105037 056413 CLRB $LFLG ;:CLEAR LOG FLAG
8502 056400 105037 056412 CLRB $MFLG ;:CLEAR MESSAGE FLAG
8503 056404 012601 MOV (SP)+,R1 ;:POP STACK INTO R1
8504 056406 012600 MOV (SP)+,R0 ;:POP STACK INTO R0
8505 056410 000207 RTS PC ;:RETURN
8506 056412 000 $MFLG: .BYTE 0 ;:MESSG. FLAG
8507 056413 000 $LFLG: .BYTE 0 ;:LOG FLAG
```



8508 056414 000  
8509 056416  
8510 000200  
8511 000001  
8512 000100  
8513 000040  
8514  
8515  
8516  
8517  
8518  
8519  
8520  
8521  
8522  
8523  
8524  
8525  
8526  
8527  
8528 056416  
8529 056416 104407  
8530 056420 105237 001103  
8531 056424 001775  
8532 056426 013777 001102 122506  
8533 056434 032777 002000 122476  
8534 056442 001402  
8535 056444 104401 001204  
8536 056450 005237 001112  
8537 056454 011637 001116  
8538 056460 162737 000002 001116  
8539 056466 117737 122424 001114  
8540 056474 032777 020000 122436  
8541 056502 001004  
8542 056504 004737 056616  
8543 056510 104401 001211  
8544 056514  
8545 056514 122737 000001 001234  
8546 056522 001007  
8547 056524 113737 001114 056536  
8548 056532 004737 056166  
8549 056536 000  
8550 056537 000  
8551 056540 000777  
8552 056542 005777 122372  
8553 056546 100002  
8554 056550 000000  
8555 056552 104407  
8556 056554 032777 001000 122356  
8557 056562 001402  
8558 056564 013716 001110  
8559 056570 005737 001202  
8560 056574 001402  
8561 056576 013716 001202  
8562 056602  
8563 056602 022737 055244 000042

\$FFLG: .BYTE 0 ;;FATAL FLAG  
.EVEN  
APTSIZE=200  
APTENV=001  
APTSPool=100  
APTCSUP=040  
.SBTTL ERROR HANDLER ROUTINE  
\*\*\*\*\*  
\*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,  
\*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL  
\*AND GO TO TYPERR ON ERROR  
\*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
\*SW15=1 HALT ON ERROR  
\*SW13=1 INHIBIT ERROR TYPEOUTS  
\*SW10=1 BELL ON ERROR  
\*SW09=1 LOOP ON ERROR  
\*CALL  
\* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER  
\$ERROR:  
7\$: CKSWR ;;TEST FOR CHANGE IN SOFT-SWR  
INCB \$ERFLG ;;SET THE ERROR FLAG  
BEQ 7\$ ;;DON'T LET THE FLAG GO TO ZERO  
MOV \$TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG  
BIT #BIT10,@SWR ;;BELL ON ERROR?  
BEQ 1\$ ;;NO - SKIP  
TYPE \$BELL ;;RING BELL  
1\$: INC \$ERTTL ;;COUNT THE NUMBER OF ERRORS  
MOV (SP),\$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION  
SUB #2,\$ERRPC  
MOVB @ERRPC,\$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE  
BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET  
BNE 20\$ ;;SKIP TYPEOUTS  
JSR PC,TYPERR ;;GO TO USER ERROR ROUTINE  
TYPE \$CRLF  
20\$: CMPB #APTENV,\$ENV ;;RUNNING IN APT MODE  
BNE 2\$ ;;NO,SKIP APT ERROR REPORT  
MOVB \$ITEMB,21\$ ;;SET ITEM NUMBER AS ERROR NUMBER  
JSR PC,\$ATY4 ;;REPORT FATAL ERROR TO APT  
21\$: .BYTE 0  
.BYTE 0  
22\$: BR 22\$ ;;APT ERROR LOOP  
2\$: TST @SWR ;;HALT ON ERROR  
BPL 3\$ ;;SKIP IF CONTINUE  
HALT ;;HALT ON ERROR!  
CKSWR ;;TEST FOR CHANGE IN SOFT-SWR  
3\$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?  
BEQ 4\$ ;;BR IF NO  
MOV \$LPERR,(SP) ;;FUDGE RETURN FOR LOOPING  
4\$: TST \$ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS  
BEQ 5\$ ;;BR IF NONE  
MOV \$ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE  
5\$: CMP #SENDAD,@#42 ;;ACT-11 AUTO-ACCEPT?



8564 056610 001001  
8565 056612 000000  
8566 056614  
8567 056614 000002  
8568  
8569  
8570  
8571  
8572  
8573  
8574  
8575  
8576  
8577  
8578  
8579 056616 104413  
8580 056620 113700 001114  
8581 056624 042700 177400  
8582 056630 005300  
8583 056632 006300  
8584 056634 006300  
8585 056636 006300  
8586 056640 062700 001300  
8587 056644 012037 056660  
8588 056650 001404  
8589 056652 104401 001211  
8590 056656 104401  
8591 056660 000000  
8592 056662 012037 056676  
8593 056666 001404  
8594 056670 104401 001211  
8595 056674 104401  
8596 056676 000000  
8597 056700 012001  
8598 056702 001445  
8599 056704 005004  
8600 056706 012000  
8601 056710 012002  
8602 056712 104401 001211  
8603 056716 112003  
8604 056720 105720  
8605 056722 005703  
8606 056724 001416  
8607 056726 005704  
8608 056730 001004  
8609 056732 013146  
8610 056734 104402  
8611 056736 005303  
8612 056740 001403  
8613 056742 104401 062243  
8614 056746 000771  
8615 056750 104401 001211  
8616 056754 005710  
8617 056756 001401  
8618 056760 005104  
8619 056762 005302

```

        BNE      6$          ;;BRANCH IF NO
        HALT                    ;;YES
6$:      RTI                    ;;RETURN

;*****
;SBTTL  TYPE ERROR ROUTINE
;*ENTRY JSR PC,TYPERR
;*RETURN RTS PC
;*
;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
;*THE ERROR.
;*****
TYPERR: SAVREG
        MOVB    $ITEMB,R0      ;ENTER ERROR NUMBER
        BIC     #177400,R0     ;CLEAR UNUSED BITS
        DEC     R0             ;FORM INDEX FOR ERROR TABLE
        ASL     R0
        ASL     R0
        ASL     R0
1$:     ADD     #$ERRTB,R0     ;FORM ADDRESS OF ERROR ENTRY
        MOV     (R0)+,2$      ;GET EM POINTER
        BEQ     3$            ;BRANCH IF THERE ISN'T ONE
        TYPE    , $CRLF       ;TYPE CARRIAGE RETURN LINE FEED
        TYPE    , $CRLF       ;TYPE ERROR MESSAGE (EM)
2$:     .WORD   0              ;EM POINTER GOES HERE
3$:     MOV     (R0)+,4$      ;GET DH POINTER
        BEQ     5$            ;BRANCH IF THERE ISN'T ONE
        TYPE    , $CRLF       ;TYPE CR-LF
        TYPE    , $CRLF       ;TYPE DATA HEADER
4$:     .WORD   0              ;DH POINTER GOES HERE
5$:     MOV     (R0)+,R1      ;GET DT POINTER
        BEQ     20$           ;BRANCH IF THERE ARE NONE
        CLR     R4            ;RESET INDENT SWITCH
        MOV     (R0)+,R0      ;GET DF POINTER
        MOV     (R0)+,R2      ;STORE NUMBER OF DH'S
        TYPE    , $CRLF       ;TYPE <CR><LF>
10$:    MOVB    (R0)+,R3      ;GET & STORE NUMBER OF DATA WORDS
        TSTB   (R0)+         ;BUMP PAST FORMAT WORD
        TST    R3            ;TEST IF ANY DATA FOR THIS HEADER
        BEQ    14$           ;NO - SKIP DATA PRINT
        TST    R4            ;CHECK IF INDENT WORDS
        BNE    12$           ;YES, GO INDENT
11$:    MOV     @ (R1)+,-(SP) ;PUT FIRST DATA WORD ON STACK
        TYPOC                    ;TYPE IT
        DEC    R3            ;MORE DATA WORDS
        BEQ    13$           ;NO-BRANCH
12$:    TYPE    ,SPACE2       ;TYPE SEPARATORS
        BR     11$           ;LOOP
13$:    TYPE    , $CRLF       ;TYPE <CR><LF>
        TST    (R0)          ;CHECK IF NEXT HEADER AVAILIBLE
        BEQ    14$           ;NO, DO NOT CHANGE INDENT
        COM    R4            ;CHANGE INDENT
14$:    DEC     R2            ;MORE DH'S?

```



```
8620 056764 003414          BLE      20$          ;NO-BRANCH
8621 056766 012037 057006   15$:    MOV      (R0)+,18$ ;GET NEXT DH POINTER
8622 056772 001751          BEQ      10$          ;IF NO HEADER GET DATA
8623 056774 005704          TST      R4           ;INDENT?
8624 056776 001402          BEQ      17$          ;NO-BRANCH
8625 057000 104401 062243     TYPE     ,SPACE2     ;INDENT
8626 057004 104401          17$:    TYPE     ;TYPE DH
8627 057006 000000          18$:    .WORD    0         ;DH POINTER GOES HERE
8628 057010 104401 001211     TYPE     ,SCLRF      ;
8629 057014 000740          BR       10$          ;LOOP
8630 057016 104414          20$:    RESREG
8631 057020 005237 002006     INC      ERRCNT      ;INCREMENT ERROR COUNT
8632 057024 032777 010000 122106  BIT      #SW12,@SWR   ;CHECK IF ABORT AFTER 20 ERRORS
8633 057032 001421          BEQ      25$          ;NO, TYPE OUT ERROR
8634 057034 022737 000024 002006  CMP      #20,ERRCNT  ;CHECK IF ERROR THRESHOLD EXCEEDED
8635 057042 103015          BHIS    25$          ;NO, PRINT ERROR
8636 057044 104401 062246     TYPE     ,ABORT      ;TYPE 'PROGRAM HAS BEEN ABORTED BECAUSE
8637                                ; ERROR THRESHOLD EXCEEDED'
8638 057050 005737 000042     TST      42           ;CHECK IF IN CHAIN MODE
8639 057054 001407          BEQ      30$          ;NO, HALT
8640 057056 012737 000001 055102  MOV      #1,$EOPCT   ;FORCE END OF PASS COUNT TO ONE FOR ABORT
8641 057064 012706 001100          MOV      #STACK,SP  ;INITIALIZE STACK
8642 057070 000137 055054          JMP      $EOP        ;BRING IN NEXT PROGRAM IN CHAIN
8643 057074 000000          30$:    HALT
8644 057076 000207          25$:    RTS      PC
8645                                .SBTTL  TYPE ROUTINE
```

```
8646
8647
8648 ::*****
8649 ::ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
8650 ::THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
8651 ::NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
8652 ::NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
8653 ::NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
8654 ::
8655 ::CALL:
8656 ::*1) USING A TRAP INSTRUCTION
8657 ::* TYPE ,MESADR ;:MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
8658 ::OR
8659 ::* TYPE
8660 ::* MESADR
8661 ::*
```

```
8662 057100 105737 001157     $TYPE: TSTB    $TPFLG  ;:IS THERE A TERMINAL?
8663 057104 100002          BPL      1$          ;:BR IF YES
8664 057106 000000          HALT
8665 057110 000430          BR       3$          ;:HALT HERE IF NO TERMINAL
8666 057112 010046          1$:    MOV      R0,-(SP) ;:LEAVE
8667 057114 017600 000002     MOV      @2(SP),R0   ;:SAVE R0
8668 057120 122737 000001 001234  CMPB    #APTENV,$ENV ;:GET ADDRESS OF ASCIZ STRING
8669 057126 001011          BNE     62$          ;:RUNNING IN APT MODE
8670 057130 132737 000100 001235  BITB    #APTPOOL,$ENVM ;:NO,GO CHECK FOR APT CONSOLE
8671 057136 001405          BEQ     62$          ;:SPOOL MESSAGE TO APT
8672 057140 010037 057150     MOV      R0,61$     ;:NO,GO CHECK FOR CONSOLE
8673 057144 004737 056156     JSR     PC,$ATY3    ;:SETUP MESSAGE ADDRESS FOR APT
8674 057150 000000          61$:    .WORD    0         ;:SPOOL MESSAGE TO APT
8675 057152 132737 000040 001235  62$:    BITB    #APTCSUP,$ENVM ;:MESSAGE ADDRESS
;:APT CONSOLE SUPPRESSED
```







```

8732 057406 001003          BNE      1$          ;;BRANCH IF NO
8733 057410 105037 057430  CLR      $CHARCNT  ;;YES--CLEAR CHARACTER COUNT
8734 057414 000406          BR       $TYPEX     ;;EXIT
8735 057416 122766 000012 000002 1$:  CMPB   #LF,2(SP)   ;;IS CHARACTER A LINE FEED?
8736 057424 001402          BEQ     $TYPEX     ;;BRANCH IF YES
8737 057426 105227          INCB   (PC)+       ;;COUNT THE CHARACTER
8738 057430 000000          $CHARCNT: .WORD 0  ;;CHARACTER COUNT STORAGE
8739 057432 000207          $TYPEX: RTS      PC

```

```

8740
8741          .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
8742
8743

```

```

8744          ;;*****
8745          ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
8746          ;;OCTAL (ASCII) NUMBER AND TYPE IT.
8747          ;;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
8748          ;;CALL:
8749          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
8750          ;;      TYPOS          ;;CALL FOR TYPEOUT
8751          ;;      .BYTE  N          ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
8752          ;;      .BYTE  M          ;;M=1 OR 0
8753          ;;                                  ;;1=TYPE LEADING ZEROS
8754          ;;                                  ;;0=SUPPRESS LEADING ZEROS

```

```

8755          ;;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
8756          ;;$TYPOS OR $TYPOC
8757          ;;CALL:

```

```

8758          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
8759          ;;      TYPON          ;;CALL FOR TYPEOUT
8760

```

```

8761          ;;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
8762          ;;CALL:

```

```

8763          ;;      MOV      NUM,-(SP)          ;;NUMBER TO BE TYPED
8764          ;;      TYPOC          ;;CALL FOR TYPEOUT
8765

```

```

8766 057434 017646 000000          $TYPOS: MOV     @(SP),-(SP)  ;;PICKUP THE MODE
8767 057440 116637 000001 057657  MOV     1(SP),$OFILL  ;;LOAD ZERO FILL SWITCH
8768 057446 112637 057661          MOV     (SP)+,$OMODE+1 ;;NUMBER OF DIGITS TO TYPE
8769 057452 062716 000002          ADD     #2,(SP)      ;;ADJUST RETURN ADDRESS
8770 057456 000406          BR      $TYPON
8771 057460 112737 000001 057657  $TYPOC: MOV     #1,$OFILL  ;;SET THE ZERO FILL SWITCH
8772 057466 112737 000006 057661  MOV     #6,$OMODE+1  ;;SET FOR SIX(6) DIGITS
8773 057474 112737 000005 057656  $TYPON: MOV     #5,$OCNT  ;;SET THE ITERATION COUNT
8774 057502 010346          MOV     R3,-(SP)    ;;SAVE R3
8775 057504 010446          MOV     R4,-(SP)    ;;SAVE R4
8776 057506 010546          MOV     R5,-(SP)    ;;SAVE R5
8777 057510 113704 057661          MOV     $OMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
8778 057514 005404          NEG     R4
8779 057516 062704 000006          ADD     #6,R4       ;;SUBTRACT IT FOR MAX. ALLOWED
8780 057522 110437 057660          MOV     R4,$OMODE  ;;SAVE IT FOR USE
8781 057526 113704 057657          MOV     $OFILL,R4  ;;GET THE ZERO FILL SWITCH
8782 057532 016605 000012          MOV     12(SP),R5  ;;PICKUP THE INPUT NUMBER
8783 057536 005003          CLR     R3         ;;CLEAR THE OUTPUT WORD
8784 057540 006105          1$:  ROL     R5       ;;ROTATE MSB INTO 'C'
8785 057542 000404          BR      3$         ;;GO DO MSB
8786 057544 006105          2$:  ROL     R5       ;;FORM THIS DIGIT
8787 057546 006105          ROL     R5

```



```

8788 057550 006105          ROL      R5
8789 057552 010503          MOV      R5,R3
8790 057554 006103          3$:    ROL      R3          ;;GET LSB OF THIS DIGIT
8791 057556 105337 057660  DEC8     $OMODE      ;;TYPE THIS DIGIT?
8792 057562 100016          BPL      7$          ;;BR IF NO
8793 057564 042703 177770  BIC      #177770,R3  ;;GET RID OF JUNK
8794 057570 001002          BNE      4$          ;;TEST FOR 0
8795 057572 005704          TST      R4          ;;SUPPRESS THIS 0?
8796 057574 001403          BEQ      5$          ;;BR IF YES
8797 057576 005204          4$:    INC      R4          ;;DON'T SUPPRESS ANYMORE 0'S
8798 057600 052703 000060  BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
8799 057604 052703 000040  5$:    BIS      #' ,R3  ;;MAKE ASCII IF NOT ALREADY
8800 057610 110337 057654  MOV8     R3,8$       ;;SAVE FOR TYPING
8801 057614 104401 057654  TYPE     ,8$         ;;GO TYPE THIS DIGIT
8802 057620 105337 057656  7$:    DEC8     $OCNT   ;;COUNT BY 1
8803 057624 003347          BGT      2$          ;;BR IF MORE TO DO
8804 057626 002402          BLT      6$          ;;BR IF DONE
8805 057630 005204          INC      R4          ;;INSURE LAST DIGIT ISN'T A BLANK
8806 057632 000744          BR       2$          ;;GO DO THE LAST DIGIT
8807 057634 012605          6$:    MOV      (SP)+,R5  ;;RESTORE R5
8808 057636 012604          MOV      (SP)+,R4  ;;RESTORE R4
8809 057640 012603          MOV      (SP)+,R3  ;;RESTORE R3
8810 057642 016666 000002 000004  MOV      2(SP),4(SP) ;;SET THE STACK FOR RETURNING
8811 057650 012616          MOV      (SP)+,(SP)
8812 057652 000002          RTI                     ;;RETURN
8813 057654 000          8$:    .BYTE    0          ;;STORAGE FOR ASCII DIGIT
8814 057655 000          .BYTE    0          ;;TERMINATOR FOR TYPE ROUTINE
8815 057656 000          $OCNT:  .BYTE    0          ;;OCTAL DIGIT COUNTER
8816 057657 000          $OFILL: .BYTE    0          ;;ZERO FILL SWITCH
8817 057660 000000          $OMODE: .WORD    0          ;;NUMBER OF DIGITS TO TYPE
8818          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
8819
8820          ;;*****
8821          ;;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
8822          ;;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
8823          ;;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
8824          ;;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
8825          ;;*REPLACED WITH SPACES.
8826          ;;*CALL:
8827          ;;*      MOV      NUM,-(SP)          ;;PUT THE BINARY NUMBER ON THE STACK
8828          ;;*      TYPDS          ;;GO TO THE ROUTINE
8829
8830          $TYPDS:
8831          MOV      R0,-(SP)          ;;PUSH R0 ON STACK
8832          MOV      R1,-(SP)          ;;PUSH R1 ON STACK
8833          MOV      R2,-(SP)          ;;PUSH R2 ON STACK
8834          MOV      R3,-(SP)          ;;PUSH R3 ON STACK
8835          MOV      R5,-(SP)          ;;PUSH R5 ON STACK
8836          MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
8837          MOV      20(SP),R5        ;;GET THE INPUT NUMBER
8838          BPL      1$          ;;BR IF INPUT IS POS.
8839          NEG      R5              ;;MAKE THE BINARY NUMBER POS.
8840          MOV8     #'-,1(SP)        ;;MAKE THE ASCII NUMBER NEG.
8841          8841 057716 005000          1$:    CLR      R0          ;;ZERO THE CONSTANTS INDEX
8842          MOV      #$DBLK,R3        ;;SETUP THE OUTPUT POINTER
8843          MOV8     #' ,(R3)+        ;;SET THE FIRST CHARACTER TO A BLANK

```



```
8844 057730 005002      2$: CLR R2          ;;CLEAR THE BCD NUMBER
8845 057732 016001 060066  MOV $DTBL(R0),R1    ;;GET THE CONSTANT
8846 057736 160105      3$: SUB R1,R5        ;;FORM THIS BCD DIGIT
8847 057740 002402      BLT 4$             ;;BR IF DONE
8848 057742 005202      INC R2             ;;INCREASE THE BCD DIGIT BY 1
8849 057744 000774      BR 3$
8850 057746 060105      4$: ADD R1,R5        ;;ADD BACK THE CONSTANT
8851 057750 005702      TST R2           ;;CHECK IF BCD DIGIT=0
8852 057752 001002      RNE 5$          ;;FALL THROUGH IF 0
8853 057754 105716      TSTB (SP)        ;;STILL DOING LEADING 0'S?
8854 057756 100407      BMI 7$          ;;BR IF YES
8855 057760 106316      5$: ASLB (SP)      ;;MSD?
8856 057762 103003      BCC 6$          ;;BR IF NO
8857 057764 116663 000001 177777  MOVB 1(SP),-1(R3)  ;;YES--SET THE SIGN
8858 057772 052702 000060  6$: BIS #'0,R2     ;;MAKE THE BCD DIGIT ASCII
8859 057776 052702 000040  7$: BIS #',R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
8860 060002 110223      MOVB R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
8861 060004 005720      TST (R0)+       ;;JUST INCREMENTING
8862 060006 020027 000010  CMP R0,#10      ;;CHECK THE TABLE INDEX
8863 060012 002746      BLT 2$          ;;GO DO THE NEXT DIGIT
8864 060014 003002      BGT 8$          ;;GO TO EXIT
8865 060016 010502      MOV R5,R2       ;;GET THE LSD
8866 060020 000764      BR 6$          ;;GO CHANGE TO ASCII
8867 060022 105726      8$: TSTB (SP)+    ;;WAS THE LSD THE FIRST NON-ZERO?
8868 060024 100003      BPL 9$          ;;BR IF NO
8869 060026 116663 177777 177776  MOVB -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
8870 060034 105013      9$: CLRB (R3)     ;;SET THE TERMINATOR
8871 060036 012605      MOV (SP)+,R5    ;;POP STACK INTO R5
8872 060040 012603      MOV (SP)+,R3    ;;POP STACK INTO R3
8873 060042 012602      MOV (SP)+,R2    ;;POP STACK INTO R2
8874 060044 012601      MOV (SP)+,R1    ;;POP STACK INTO R1
8875 060046 012600      MOV (SP)+,R0    ;;POP STACK INTO R0
8876 060050 104401 060076  TYPE $DBLK      ;;NOW TYPE THE NUMBER
8877 060054 016666 000002 000004  MOV 2(SP),4(SP)  ;;ADJUST THE STACK
8878 060062 012616      MOV (SP)+,(SP)
8879 050064 000002      RTI             ;;RETURN TO USER
8880 060066 023420      $DTBL: 10000.
8881 060070 001750      1000.
8882 060072 000144      100.
8883 060074 000012      10.
8884 060076 000004      $DBLK: .BLKW 4
8885      .SBTTL TTY INPUT ROUTINE
8886
8887      ;:*****
8888      .ENABL LSB
8889
8890      ;:*****
8891      ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
8892      ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
8893      ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
8894      ;*WHEN OPERATING IN TTY FLAG MODE.
8895 060106 022737 000176 001140 $CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?
8896 060114 001074      BNE 15$        ;;BRANCH IF NO
8897 060116 105777 121022      TSTB @TKS     ;;CHAR THERE?
8898 060122 100071      BPL 15$        ;;IF NO, DON'T WAIT AROUND
8899 060124 117746 121016      MOVB @TKB,-(SP) ;;SAVE THE CHAR
```



8900	060130	042716	177600		BIC	#^C177,(SP)	::STRIP-OFF THE ASCII
8901	060134	022726	000007		CMP	#7,(SP)+	::IS IT A CONTROL G?
8902	060140	001062			BNE	15\$	::NO, RETURN TO USER
8903	060142	123727	001134	000001	CMPB	\$AUTOB,#1	::ARE WE RUNNING IN AUTO-MODE?
8904	060150	001456			BEQ	15\$	::BRANCH IF YES
8905							
8906	060152	104401	060771		TYPE	.\$CNTLG	::ECHO THE CONTROL-G (^G)
8907	060156	104401	060776	\$GTSWR:	TYPE	.\$MSWR	::TYPE CURRENT CONTENTS
8908	060162	013746	000176		MOV	SWREG,-(SP)	::SAVE SWREG FOR TYPEOUT
8909	060166	104402			TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
8910	060170	104401	061007		TYPE	.\$MNEW	::PROMPT FOR NEW SWR
8911	060174	005046		19\$:	CLR	-(SP)	::CLEAR COUNTER
8912	060176	005046			CLR	-(SP)	::THE NEW SWR
8913	060200	105777	120740	7\$:	TSTB	@\$TKS	::CHAR THERE?
8914	060204	100375			BPL	7\$	::IF NOT TRY AGAIN
8915							
8916	060206	117746	120734		MOVB	@\$TKB,-(SP)	::PICK UP CHAR
8917	060212	042716	177600		BIC	#^C177,(SP)	::MAKE IT 7-BIT ASCII
8918							
8919							
8920							
8921	060216	021627	000025	9\$:	CMP	(SP),#25	::IS IT A CONTROL-U?
8922	060222	001005			BNE	10\$	::BRANCH IF NOT
8923	060224	104401	060764		TYPE	.\$CNTLU	::YES, ECHO CONTROL-U (^U)
8924	060230	062706	000006	20\$:	ADD	#6,SP	::IGNORE PREVIOUS INPUT
8925	060234	000757			BR	19\$	::LET'S TRY IT AGAIN
8926							
8927							
8928	060236	021627	000015	10\$:	CMP	(SP),#15	::IS IT A <CR>?
8929	060242	001022			BNE	16\$	::BRANCH IF NO
8930	060244	005766	000004		TST	4(SP)	::YES, IS IT THE FIRST CHAR?
8931	060250	001403			BEQ	11\$	::BRANCH IF YES
8932	060252	016677	000002	120660	MOV	2(SP),@SWR	::SAVE NEW SWR
8933	060260	062706	000006	11\$:	ADD	#6,SP	::CLEAR UP STACK
8934	060264	104401	001211	14\$:	TYPE	.\$CRLF	::ECHO <CR> AND <LF>
8935	060270	123727	001135	000001	CMPB	\$INTAG,#1	::RE-ENABLE TTY KBD INTERRUPTS?
8936	060276	001003			BNE	15\$	::BRANCH IF NOT
8937	060300	012777	000100	120636	MOV	#100,@\$TKS	::RE-ENABLE TTY KBD INTERRUPTS
8938	060306	000002		15\$:	RTI		::RETURN
8939	060310	004737	057312	16\$:	JSR	PC,\$TYPEC	::ECHO CHAR
8940	060314	021627	000060		CMP	(SP),#60	::CHAR < 0?
8941	060320	002420			BLT	18\$	::BRANCH IF YES
8942	060322	021627	000067		CMP	(SP),#67	::CHAR > 7?
8943	060326	003015			BGT	18\$	::BRANCH IF YES
8944	060330	042726	000060		BIC	#60,(SP)+	::STRIP-OFF ASCII
8945	060334	005766	000002		TST	2(SP)	::IS THIS THE FIRST CHAR
8946	060340	001403			BEQ	17\$	::BRANCH IF YES
8947	060342	006316			ASL	(SP)	::NO, SHIFT PRESENT
8948	060344	006316			ASL	(SP)	::CHAR OVER TO MAKE
8949	060346	006316			ASL	(SP)	::ROOM FOR NEW ONE.
8950	060350	005266	000002	17\$:	INC	2(SP)	::KEEP COUNT OF CHAR
8951	060354	056616	177776		BIS	-2(SP),(SP)	::SET IN NEW CHAR
8952	060360	000707			BR	7\$	::GET THE NEXT ONE
8953	060362	104401	001210	18\$:	TYPE	.\$QUES	::TYPE ?<CR><LF>
8954	060366	000720			BR	20\$	::SIMULATE CONTROL-U
8955					.DSABL	LSB	



```
8956  
8957  
8958  
8959  
8960  
8961  
8962  
8963  
8964  
8965  
8966 060370 011646  
8967 060372 016666 000004 000002  
8968 060400 105777 120540  
8969 060404 100375  
8970 060406 117766 120534 000004  
8971 060414 042766 177600 000004  
8972 060422 026627 000004 000023  
8973 060430 001013  
8974 060432 105777 120506  
8975 060436 100375  
8976 060440 117746 120502  
8977 060444 042716 177600  
8978 060450 022627 000021  
8979 060454 001366  
8980 060456 000750  
8981 060460 026627 000004 000021  
8982 060466 001744  
8983 060470 026627 000004 000140  
8984 060476 002407  
8985 060500 026627 000004 000175  
8986 060506 003003  
8987 060510 042766 000040 000004  
8988 060516 000002  
8989  
8990  
8991  
8992  
8993  
8994  
8995  
8996 060520 010346  
8997 060522 005046  
8998 060524 012703 060754  
8999 060530 022703 060764  
9000 060534 101456  
9001 060536 104410  
9002 060540 112613  
9003 060542 122713 000177  
9004 060546 001022  
9005 060550 005716  
9006 060552 001007  
9007 060554 112737 000134 060752  
9008 060562 104401 060752  
9009 060566 012716 177777  
9010 060572 005303  
9011 060574 020327 060754
```

```
*****  
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY  
*CALL:  
* RDCHR ;:INPUT A SINGLE CHARACTER FROM THE TTY  
* RETURN HERE ;:CHARACTER IS ON THE STACK  
* ;:WITH PARITY BIT STRIPPED OFF  
*  
$RDCHR: MOV (SP),-(SP) ;:PUSH DOWN THE PC  
MOV 4(SP),2(SP) ;:SAVE THE PS  
1$: TSTB @ $TKS ;:WAIT FOR  
BPL 1$ ;:A CHARACTER  
MOV 4(SP),4(SP) ;:READ THE TTY  
BIC #^C<177>,4(SP) ;:GET RID OF JUNK IF ANY  
CMP 4(SP),#23 ;:IS IT A CONTROL-S?  
BNE 3$ ;:BRANCH IF NO  
2$: TSTB @ $TKS ;:WAIT FOR A CHARACTER  
BPL 2$ ;:LOOP UNTIL ITS THERE  
MOV @ $TKB, -(SP) ;:GET CHARACTER  
BIC #^C177,(SP) ;:MAKE IT 7-BIT ASCII  
CMP (SP)+,#21 ;:IS IT A CONTROL-Q?  
BNE 2$ ;:IF NOT DISCARD IT  
BR 1$ ;:YES, RESUME  
3$: CMP 4(SP),#$XON ;:IS IT A RANDOM XON? ;:RAN001  
BEQ 1$ ;:BRANCH IF YES ;:RAN001  
CMP 4(SP),#140 ;:IS IT UPPER CASE?  
BLT 4$ ;:BRANCH IF YES  
CMP 4(SP),#175 ;:IS IT A SPECIAL CHAR?  
BGT 4$ ;:BRANCH IF YES  
BIC #40,4(SP) ;:MAKE IT UPPER CASE  
4$: RTI ;:GO BACK TO USER  
*****  
*THIS ROUTINE WILL INPUT A STRING FROM THE TTY  
*CALL:  
* RDLIN ;:INPUT A STRING FROM THE TTY  
* RETURN HERE ;:ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK  
* ;:TERMINATOR WILL BE A BYTE OF ALL 0'S  
*  
$RDLIN: MOV R3, -(SP) ;:SAVE R3  
CLR -(SP) ;:CLEAR THE RUBOUT KEY  
1$: MOV # $TTYIN, R3 ;:GET ADDRESS  
2$: CMP # $TTYIN+8., R3 ;:BUFFER FULL?  
BLOS 4$ ;:BR IF YES  
RDCHR ;:GO READ ONE CHARACTER FROM THE TTY  
MOV (SP)+, (R3) ;:GET CHARACTER  
10$: CMPB #177, (R3) ;:IS IT A RUBOUT  
BNE 5$ ;:BR IF NO  
TST (SP) ;:IS THIS THE FIRST RUBOUT?  
BNE 6$ ;:BR IF NO  
MOV #'\, 9$ ;:TYPE A BACK SLASH  
TYPE ,9$ ;:TYPE A BACK SLASH  
6$: MOV #-1, (SP) ;:SET THE RUBOUT KEY  
DEC R3 ;:BACKUP BY ONE  
CMP R3, # $TTYIN ;:STACK EMPTY?
```







```
9068 061020 011646          $RDOCT: MOV      (SP),-(SP)      ;;PROVIDE SPACE FOR THE
9069 061022 016666 000004 000002 MOV      4(SP),2(SP)      ;;INPUT NUMBER
9070 061030 010046          MOV      R0,-(SP)        ;;PUSH R0 ON STACK
9071 061032 010146          MOV      R1,-(SP)        ;;PUSH R1 ON STACK
9072 061034 010246          MOV      R2,-(SP)        ;;PUSH R2 ON STACK
9073 061036 104411          1$:  RDLIN          ;;READ AN ASCII LINE
9074 061040 012600          MOV      (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
9075 061042 010037 061146  MOV      R0,5$          ;;AND SAVE IT
9076 061046 005001          CLR      R1              ;;CLEAR DATA WORD
9077 061050 005002          CLR      R2
9078 061052 112046          2$:  MOVVB      (R0)+,-(SP)  ;;PICKUP THIS CHARACTER
9079 061054 001420          BEQ      3$              ;;IF ZERO GET OUT
9080 061056 122716 000060  CMPB     #'0,(SP)        ;;MAKE SURE THIS CHARACTER
9081 061062 003026          BGT      4$              ;;IS AN OCTAL DIGIT
9082 061064 122716 000067  CMPB     #'7,(SP)
9083 061070 002423          BLT      4$
9084 061072 006301          ASL      R1              ;;*2
9085 061074 006102          ROL      R2
9086 061076 006301          ASL      R1              ;;*4
9087 061100 006102          ROL      R2
9088 061102 006301          ASL      R1              ;;*8
9089 061104 006102          ROL      R2
9090 061106 042716 177770  BIC      #'C7,(SP)      ;;STRIP THE ASCII JUNK
9091 061112 062601          ADD      (SP)+,R1        ;;ADD IN THIS DIGIT
9092 061114 000756          BR       2$              ;;LOOP
9093 061116 005726          3$:  TST      (SP)+        ;;CLEAN TERMINATOR FROM STACK
9094 061120 010166 000012  MOV      R1,12(SP)      ;;SAVE THE RESULT
9095 061124 010237 061156  MOV      R2,$HIOCT
9096 061130 012602          MOV      (SP)+,R2        ;;POP STACK INTO R2
9097 061132 012601          MOV      (SP)+,R1        ;;POP STACK INTO R1
9098 061134 012600          MOV      (SP)+,R0        ;;POP STACK INTO R0
9099 061136 000002          RTI                      ;;RETURN
9100 061140 005726          4$:  TST      (SP)+        ;;CLEAN PARTIAL FROM STACK
9101 061142 105010          CLR      (R0)           ;;SET A TERMINATOR
9102 061144 104401          TYPE          ;;TYPE UP THRU THE BAD CHAR.
9103 061146 000000          5$:  .WORD     0
9104 061150 104401 001210  TYPE     $QUES          ;;'"?' 'CR' & 'LF'
9105 061154 000730          BR       1$              ;;TRY AGAIN
9106 061156 000000  $HIOCT: .WORD     0      ;;HIGH ORDER BITS GO HERE
9107          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
9108
9109          ;*****
9110          ;*SAVE R0-R5
9111          ;*CALL:
9112          ;* SAVREG
9113          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
9114          ;*
9115          ;*TOP---(+16)
9116          ;* +2---(+18)
9117          ;* +4---R5
9118          ;* +6---R4
9119          ;* +8---R3
9120          ;*+10---R2
9121          ;*+12---R1
9122          ;*+14---R0
9123
```



9124 061160  
9125 061160 010046  
9126 061162 010146  
9127 061164 010246  
9128 061166 010346  
9129 061170 010446  
9130 061172 010546  
9131 061174 016646 000022  
9132 061200 016646 000022  
9133 061204 016646 000022  
9134 061210 016646 000022  
9135 061214 000002  
9136  
9137  
9138  
9139  
9140 061216  
9141 061216 012666 000022  
9142 061222 012666 000022  
9143 061226 012666 000022  
9144 061232 012666 000022  
9145 061236 012605  
9146 061240 012604  
9147 061242 012603  
9148 061244 012602  
9149 061246 012601  
9150 061250 012600  
9151 061252 000002  
9152  
9153  
9154  
9155  
9156  
9157 061254 017737 117660 002026  
9158 061262 012737 061302 000024  
9159 061270 012737 000340 000026  
9160 061276 000000  
9161 061300 000776  
9162  
9163  
9164  
9165 061302 005037 061372  
9166 061306 012737 000144 061374  
9167 061314 005237 061372  
9168 061320 001375  
9169 061322 005337 061374  
9170 061326 001372  
9171 061330 012737 061254 000024  
9172 061336 012737 000340 000026  
9173 061344 012706 001100  
9174 061350 104401 061376  
9175 061354 004737 055264  
9176 061360 013777 002026 117552  
9177 061366 000177 117514  
9178  
9179 061372 000000 000000

```

$SAVREG:
MOV R0,-(SP)      ;;PUSH R0 ON STACK
MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV R2,-(SP)      ;;PUSH R2 ON STACK
MOV R3,-(SP)      ;;PUSH R3 ON STACK
MOV R4,-(SP)      ;;PUSH R4 ON STACK
MOV R5,-(SP)      ;;PUSH R5 ON STACK
MOV 22(SP),-(SP)  ;;SAVE PS OF MAIN FLOW
MOV 22(SP),-(SP)  ;;SAVE PC OF MAIN FLOW
MOV 22(SP),-(SP)  ;;SAVE PS OF CALL
MOV 22(SP),-(SP)  ;;SAVE PC OF CALL
RTI

;*RESTORE R0-R5
;*CALL:
;* RESREG
$RESREG:
MOV (SP)+,22(SP)  ;;RESTORE PC OF CALL
MOV (SP)+,22(SP)  ;;RESTORE PS OF CALL
MOV (SP)+,22(SP)  ;;RESTORE PC OF MAIN FLOW
MOV (SP)+,22(SP)  ;;RESTORE PS OF MAIN FLOW
MOV (SP)+,R5      ;;POP STACK INTO R5
MOV (SP)+,R4      ;;POP STACK INTO R4
MOV (SP)+,R3      ;;POP STACK INTO R3
MOV (SP)+,R2      ;;POP STACK INTO R2
MOV (SP)+,R1      ;;POP STACK INTO R1
MOV (SP)+,R0      ;;POP STACK INTO R0
RTI

.SBTTL POWER DOWN AND UP ROUTINES

*****
:POWER DOWN ROUTINE
$PWRDN: MOV @SWR,SAVSWR  ;;SAVE SWITCH REGISTER
MOV #PWRUP,PWRVEC  ;;SET UP VECTOR
MOV #PR7,PWRVEC+2
HALT
BR .-2 ;HANG UP

*****
:POWER UP ROUTINE
$PWRUP: CLR $PWRCT      ;;LOAD WAIT COUNT
MOV #100,$PWRCT+2
1$: INC $PWRCT          ;;WAIT FOR TELETYPE
BNE 1$
DEC $PWRCT+2
BNE 1$
MOV #PWRDN,PWRVEC  ;;SET UP FOR POWER DOWN VECTOR
MOV #PR7,PWRVEC+2
MOV #STACK,SP      ;;FORCE STACK
TYPE $POWER        ;;TYPE POWER
JSR PC,CHKPAR      ;;REINITIALIZE MEMORY CHECK ENABLE
MOV SAVSWR,@SWR    ;;RESTORE SWITCH REGISTER
JMP @SLPADR        ;;GO BACK TO LAST TEST

$PWRCT: .WORD 0,0  ;;TELETYPE TIME OUT
```



9180 061376 047520 042'27 U00122  
9181  
9182  
9183  
9184  
9185  
9186  
9187  
9188  
9189  
9190 061404 010046  
9191 061406 016600 000002  
9192 061412 005740  
9193 061414 111000  
9194 061416 006300  
9195 061420 016000 061440  
9196 061424 000200  
9197  
9198  
9199  
9200  
9201 061426 011646  
9202 061430 016666 000004 000002  
9203 061436 000002  
9204  
9205  
9206  
9207  
9208  
9209  
9210  
9211  
9212 061440 061426  
9213 061442 057100  
9214 061444 057460  
9215 061446 057434  
9216 061450 057474  
9217 061452 057662  
9218  
9219 061454 060156  
9220  
9221 061456 060106  
9222 061460 060370  
9223 061462 060520  
9224 061464 061020  
9225 061466 061160  
9226 061470 061216  
9227 061472 056124

\$POWER: .ASCIZ /POWER/  
.EVEN  
.SBTTL TRAP DECODER

\*\*\*\*\*  
\*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION  
\*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS  
\*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
\*GO TO THAT ROUTINE.

\$TRAP: MOV R0,-(SP) ::SAVE R0  
MOV 2(SP),R0 ::GET TRAP ADDRESS  
TST -(R0) ::BACKUP BY 2  
MOVB (R0),R0 ::GET RIGHT BYTE OF TRAP  
ASL R0 ::POSITION FOR INDEXING  
MOV \$TRPAD(R0),R0 ::INDEX TO TABLE  
RTS R0 ::GO TO ROUTINE

::THIS IS USE TO HANDLE THE "GETPRI" MACRO

\$TRAP2: MOV (SP),-(SP) ::MOVE THE PC DOWN  
MOV 4(SP),2(SP) ::MOVE THE PSW DOWN  
RTI ::RESTORE THE PSW

.SBTTL TRAP TABLE

\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
\*BY THE "TRAP" INSTRUCTION.

	ROUTINE		
\$TRPAD	.WORD \$TRAP2	TRAP+1(104401)	TTY TYPEOUT ROUTINE
\$TYPE	::CALL=TYPE	TRAP+2(104402)	TYPE OCTAL NUMBER (WITH LEADING ZEROS)
\$TYPOC	::CALL=TYPOC	TRAP+3(104403)	TYPE OCTAL NUMBER (NO LEADING ZEROS)
\$TYPOS	::CALL=TYPOS	TRAP+4(104404)	TYPE OCTAL NUMBER (AS PER LAST CALL)
\$TYPON	::CALL=TYPON	TRAP+5(104405)	TYPE DECIMAL NUMBER (WITH SIGN)
\$TYPDS	::CALL=TYPDS		
\$GTSWR	::CALL=GTSWR	TRAP+6(104406)	GET SOFT-SWR SETTING
\$CKSWR	::CALL=CKSWR	TRAP+7(104407)	TEST FOR CHANGE IN SOFT-SWR
\$RDCHR	::CALL=RDCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE
\$RDLIN	::CALL=RDLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE
\$RDOCT	::CALL=RDOCT	TRAP+12(104412)	READ AN OCTAL NUMBER FROM TTY
\$SAVREG	::CALL=SAVREG	TRAP+13(104413)	SAVE R0-R5 ROUTINE
\$RESREG	::CALL=RESREG	TRAP+14(104414)	RESTORE R0-R5 ROUTINE
\$SCOP1\$	::CALL=SCOP1	TRAP+15(104415)	INTERNAL LOOP ON ERROR



```
9228 .SBTTL DATA TABLE FOR ERROR PRINT OUT
9229
9230 061474 001220 002024 DT000: .WORD $TESTN,TRAPPC
9231 061500 001220 001116 001122 DT001: .WORD $TESTN,$ERRPC,$BDADR
9232 061506 001220 001116 001124 DT002: .WORD $TESTN,$ERRPC,$GDDAT,$BDDAT
9233 061514 001126
9234 061516 001220 001116 002010 DT003: .WORD $TESTN,$ERRPC,CONFIG,$GDDAT,$BDDAT
9235 061524 001124 001126
9236 061530 001220 001116 002014 DT004: .WORD $TESTN,$ERRPC,PREREG,$GDDAT,$BDDAT
9237 061536 001124 001126
9238 061542 001220 001116 DT005: .WORD $TESTN,$ERRPC
9239 061546 001220 001116 002000 DT006: .WORD $TESTN,$ERRPC,RKVEC,$BDADR
9240 061554 001122
9241 061556 001220 001116 002020 DT011: .WORD $TESTN,$ERRPC,PRIOR
9242 061564 001220 001116 001740 DT016: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2
9243 061572 001700 001750 001710
9244 061600 001220 001116 001740 DT024: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,SILCNT
9245 061606 001700 001750 001710
9246 061614 002022
9247 061616 001220 001116 001740 DT026: .WORD $TESTN,$ERRPC,E.CS1,T.CS1,E.CS2,T.CS2,SILCNT,$GDDAT,$BDDAT
9248 061624 001700 001750 001710
9249 061632 002022 001124 001126
```



				.SBTTL DATA FORMAT FOR ERROR PRINT OUT	
9250					
9251					
9252	061640	000001		DF000:	.WORD 1
9253	061642	002	000		.BYTE 2.0
9254	061644	000004		DF001:	.WORD 4 ;ERROR 1
9255	061646	000	000		.BYTE 0.0
9256	061650	062333			.WORD DH000A
9257	061652	000	000		.BYTE 0.0
9258	061654	062351			.WORD DH000B
9259	061656	002	000		.BYTE 2.0
9260	061660	062414			.WORD DH001
9261	061662	001	000		.BYTE 1.0
9262	061664	000005		DF002:	.WORD 5 ;ERROR 2
9263	061666	000	000		.BYTE 0.0
9264	061670	062333			.WORD DH000A
9265	061672	000	000		.BYTE 0.0
9266	061674	062351			.WORD DH000B
9267	061676	002	000		.BYTE 2.0
9268	061700	062433			.WORD DH002A
9269	061702	000	000		.BYTE 0.0
9270	061704	062452			.WORD DH002B
9271	061706	002	000		.BYTE 2.0
9272	061710	000005		DF003:	.WORD 5 ;ERROR 3
9273	061712	000	000		.BYTE 0.0
9274	061714	062333			.WORD DH000A
9275	061716	000	000		.BYTE 0.0
9276	061720	062351			.WORD DH000B
9277	061722	002	000		.BYTE 2.0
9278	061724	062470			.WORD DH003A
9279	061726	000	000		.BYTE 0.0
9280	061730	062517			.WORD DH003B
9281	061732	003	000		.BYTE 3.0
9282	061734	000005		DF004:	.WORD 5 ;ERROR 4
9283	061736	000	000		.BYTE 0.0
9284	061740	062333			.WORD DH000A
9285	061742	000	000		.BYTE 0.0
9286	061744	062351			.WORD DH000B
9287	061746	002	000		.BYTE 2.0
9288	061750	062545			.WORD DH004A
9289	061752	000	000		.BYTE 0.0
9290	061754	062574			.WORD DH004B
9291	061756	003	000		.BYTE 3.0
9292	061760	000003		DF005:	.WORD 3 ;ERROR 5
9293	061762	000	000		.BYTE 0.0
9294	061764	062333			.WORD DH000A
9295	061766	000	000		.BYTE 0.0
9296	061770	062351			.WORD DH000B
9297	061772	002	000		.BYTE 2.0
9298	061774	000005		DF006:	.WORD 5 ;ERROR 6
9299	061776	000	000		.BYTE 0.0
9300	062000	062333			.WORD DH000A
9301	062002	000	000		.BYTE 0.0
9302	062004	062351			.WORD DH000B
9303	062006	002	000		.BYTE 2.0
9304	062010	062622			.WORD DH006A



9305	062012	000	000		.BYTE	0.0		
9306	062014	062641			.WORD	DH006B		
9307	062016	002	000		.BYTE	2.0		
9308	062020	000004		DF011:	.WORD	4		:ERROR 11
9309	062022	000	000		.BYTE	0.0		
9310	062024	062333			.WORD	DH000A		
9311	062026	000	000		.BYTE	0.0		
9312	062030	062351			.WORD	DH000B		
9313	062032	002	000		.BYTE	2.0		
9314	062034	062660			.WORD	DH011		
9315	062036	001	000		.BYTE	1.0		
9316	062040	000005		DF016:	.WORD	5		:ERROR 16
9317	062042	000	000		.BYTE	0.0		
9318	062044	062333			.WORD	DH000A		
9319	062046	000	000		.BYTE	0.0		
9320	062050	062351			.WORD	DH000B		
9321	062052	002	000		.BYTE	2.0		
9322	062054	062622			.WORD	DH006A		
9323	062056	000	000		.BYTE	0.0		
9324	062060	062742			.WORD	DH016B		
9325	062062	004	000		.BYTE	4.0		
9326	062064	000005		DF024:	.WORD	5		:ERROR 24
9327	062066	000	000		.BYTE	0.0		
9328	062070	062333			.WORD	DH000A		
9329	062072	000	000		.BYTE	0.0		
9330	062074	062351			.WORD	DH000B		
9331	062076	002	000		.BYTE	2.0		
9332	062100	063000			.WORD	DH024A		
9333	062102	000	000		.BYTE	0.0		
9334	062104	063045			.WORD	DH024B		
9335	062106	005	000		.BYTE	5.0		
9336	062110	000005		DF026:	.WORD	5		:ERROR 26
9337	062112	000	000		.BYTE	0.0		
9338	062114	062333			.WORD	DH000A		
9339	062116	000	000		.BYTE	0.0		
9340	062120	062351			.WORD	DH000B		
9341	062122	002	000		.BYTE	2.0		
9342	062124	063112			.WORD	DH026A		
9343	062126	000	000		.BYTE	0.0		
9344	062130	063201			.WORD	DH026B		
9345	062132	007	000		.BYTE	7.0		







.SBTTL DATA HEADERS					
9373					
9374					
9375	062333	124	051505	020124	DH000A: .ASCIZ /TEST ERROR/
9376	062340	020040	042440	051122	
9377	062346	051117	000		
9378	062351	116	046525	020040	DH000B: .ASCIZ /NUM PC/
9379	062356	020040	050040	000103	
9380	062364	042524	052123	020040	DH000C: .ASCII /TEST TRAP/<15><12>
9381	062372	020040	051124	050101	
9382	062400	005015			
9383	062402	052516	020115	020040	.ASCIZ /NUM PC/
9384	062410	050040	000103		
9385	062414	047125	041111	051525	DH001: .ASCIZ /UNIBUS ADDRESS/
9386	062422	040440	042104	042522	
9387	062430	051523	000		
9388	062433	105	050130	041505	DH002A: .ASCIZ /EXPECT ACTUAL/
9389	062440	020124	040440	052103	
9390	062446	040525	000114		
9391	062452	040526	053114	020105	DH002B: .ASCIZ /VALVE VALUE/
9392	062460	020040	040526	052514	
9393	062466	000105			
9394	062470	040526	052514	020105	DH003A: .ASCIZ /VALUE EXPECT ACTUAL/
9395	062476	020040	054105	042520	
9396	062504	052103	020040	041501	
9397	062512	052524	046101	000	
9398	062517	127	044522	052124	DH003B: .ASCIZ /WRITTEN VALVE VALUE/
9399	062524	047105	053040	046101	
9400	062532	042526	020040	053040	
9401	062540	046101	042525	000	
9402	062545	120	042522	020126	DH004A: .ASCIZ /PREV EXPECT ACTUAL/
9403	062552	020040	042440	050130	
9404	062560	041505	020124	040440	
9405	062566	052103	040525	000114	
9406	062574	040526	052514	020105	DH004B: .ASCIZ /VALUE VALVE VALVE/
9407	062602	020040	040526	053114	
9408	062610	020105	020040	040526	
9409	062616	053114	000105		
9410	062622	054105	042520	052103	DH006A: .ASCIZ /EXPECT ACTUAL/
9411	062630	020040	041501	052524	
9412	062636	046101	000		
9413	062641	126	041505	047524	DH006B: .ASCIZ /VECTOR VECTOR/
9414	062646	020122	053040	041505	
9415	062654	047524	000122		
9416	062660	051120	041517	051505	DH011: .ASCIZ /PROCESSOR PRIORITY/
9417	062666	047523	020122	051120	
9418	062674	047511	044522	054524	
9419	062702	000			
9420	062703	105	050130	041505	DH016A: .ASCIZ /EXPECT ACTUAL EXPECT ACTUAL/
9421	062710	020124	040440	052103	
9422	062716	040525	020114	042440	
9423	062724	050130	041505	020124	
9424	062732	040440	052103	040525	
9425	062740	000114			
9426	062742	045522	051503	020061	DH016B: .ASCIZ /RKCS1 RKCS1 RKCS2 RKCS2/
9427	062750	020040	045522	051503	
9428	062756	020061	020040	045522	







```
9464  
9465  
9466 063266 047125 054105 042520  
9467 063274 052103 042105 046440  
9468 063302 046505 051117 020131  
9469 063310 040520 044522 054524  
9470 063316 042440 040516 046102  
9471 063324 020105 051124 050101  
9472 063332 000  
9473 063333 116 047117 042455  
9474 063340 044530 052123 047105  
9475 063346 020124 042515 047515  
9476 063354 054522 053440 042510  
9477 063362 020116 041501 042503  
9478 063370 051523 047111 020107  
9479 063376 045522 030466 020061  
9480 063404 042522 000107  
9481 063410 052101 042524 050115  
9482 063416 044524 043516 052040  
9483 063424 020117 046103 040505  
9484 063432 020122 045522 030466  
9485 063440 020061 044527 044124  
9486 063446 040440 052440 044516  
9487 063454 052502 020123 047111  
9488 063462 052111 000  
9489 063465 101 052124 046505  
9490 063472 052120 047111 020107  
9491 063500 047524 041440 042514  
9492 063506 051101 051040 033113  
9493 063514 030461 053440 052111  
9494 063522 020110 020101 047503  
9495 063530 052116 047522 046114  
9496 063536 051105 041440 042514  
9497 063544 051101 000  
9498 063547 101 052124 046505  
9499 063554 052120 047111 020107  
9500 063562 047524 053440 044522  
9501 063570 042524 041040 051525  
9502 063576 040440 042104 000  
9503 063603 101 052124 046505  
9504 063610 052120 047111 020107  
9505 063616 047524 041440 042514  
9506 063624 051101 041040 051525  
9507 063632 040440 042104 053440  
9508 063640 052111 020110 020101  
9509 063646 047125 041111 051525  
9510 063654 044440 044516 000124  
9511 063662 052101 042524 050115  
9512 063670 044524 043516 052040  
9513 063676 020117 046103 040505  
9514 063704 020122 052502 020123  
9515 063712 042101 104  
9516 063715 127 052111 020110  
9517 063722 020101 047503 052116  
9518 063730 047522 046114 051105  
9519 063736 041440 042514 051101
```

.SBTTL ERROR MESSAGES

EM000: .ASCIZ /UNEXPECTED MEMORY PARITY ENABLE TRAP/

EM1: .ASCIZ /NON-EXISTENT MEMORY WHEN ACCESSING RK611 REG/

EM2: .ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A UNIBUS INIT/

EM3: .ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A CONTROLLER CLEAR/

EM4: .ASCIZ /ATTEMPTING TO WRITE BUS ADD/

EM5: .ASCIZ /ATTEMPTING TO CLEAR BUS ADD WITH A UNIBUS INIT/

EM6: .ASCII /ATTEMPTING TO CLEAR BUS ADD/  
.ASCIZ /WITH A CONTROLLER CLEAR/



9520	063744	000			
9521	063745	101	052124	046505	EM7: .ASCIZ /ATTEMPTING TO WRITE WORD COUNT/
9522	063752	052120	047111	020107	
9523	063760	047524	053440	044522	
9524	063766	042524	053440	051117	
9525	063774	020104	047503	047125	
9526	064002	000124			
9527	064004	052101	042524	050115	EM8: .ASCIZ /ATTEMPTING TO WRITE DISK ADD/
9528	064012	044524	043516	052040	
9529	064020	020117	051127	052111	
9530	064026	020105	044504	045523	
9531	064034	040440	042104	000	
9532	064041	101	052124	046505	EM9: .ASCIZ /ATTEMPTING TO WRITE SPARE REG/
9533	064046	052120	047111	020107	
9534	064054	047524	053440	044522	
9535	064062	042524	051440	040520	
9536	064070	042522	051040	043505	
9537	064076	000			
9538	064077	101	052124	046505	EM10: .ASCIZ /ATTEMPTING TO WRITE DRIVE STATUS REG/
9539	064104	052120	047111	020107	
9540	064112	047524	053440	044522	
9541	064120	042524	042040	044522	
9542	064126	042526	051440	040524	
9543	064134	052524	020123	042522	
9544	064142	000107			
9545	064144	052101	042524	050115	EM11: .ASCIZ /ATTEMPTING TO WRITE ERROR REG/
9546	064152	044524	043516	052040	
9547	064160	020117	051127	052111	
9548	064166	020105	051105	047522	
9549	064174	020122	042522	000107	
9550	064202	052101	042524	050115	EM12: .ASCIZ /ATTEMPTING TO WRITE MAINT.REG 2/
9551	064210	044524	043516	052040	
9552	064216	020117	051127	052111	
9553	064224	020105	040515	047111	
9554	064232	027124	042522	020107	
9555	064240	000062			
9556	064242	052101	042524	050115	EM13: .ASCIZ /ATTEMPTING TO WRITE MAINT REG 3/
9557	064250	044524	043516	052040	
9558	064256	020117	051127	052111	
9559	064264	020105	040515	047111	
9560	064272	020124	042522	020107	
9561	064300	000063			
9562	064302	052101	042524	050115	EM14: .ASCIZ /ATTEMPTING TO WRITE ECC PATTERN/
9563	064310	044524	043516	052040	
9564	064316	020117	051127	052111	
9565	064324	020105	041505	020103	
9566	064332	040520	052124	051105	
9567	064340	000116			
9568	064342	052101	042524	050115	EM15: .ASCIZ /ATTEMPTING TO WRITE ECC POSITION/
9569	064350	044524	043516	052040	
9570	064356	020117	051127	052111	
9571	064364	020105	041505	020103	
9572	064372	047520	044523	044524	
9573	064400	047117	000		
9574	064403	101	052124	046505	EM16: .ASCIZ /ATTEMPTING TO WRITE CS1/
9575	064410	052120	047111	020107	



9576	064416	047524	053440	044522	
9577	064424	042524	041440	030523	
9578	064432	000			
9579	064433	101	052124	046505	EM17: .ASCIZ "ATTEMPTING TO WRITE ATTENTION SUMMARY/OFFSET"
9580	064440	052120	047111	020107	
9581	064446	047524	053440	044522	
9582	064454	042524	040440	052124	
9583	064462	047105	044524	047117	
9584	064470	051440	046525	040515	
9585	064476	054522	047457	043106	
9586	064504	042523	000124		
9587	064510	052101	042524	050115	EM18: .ASCIZ /ATTEMPTING TO WRITE CS2/
9588	064516	044524	043516	052040	
9589	064524	020117	051127	052111	
9590	064532	020105	051503	000062	
9591	064540	052101	042524	050115	EM19: .ASCIZ /ATTEMPTING TO WRITE DISK CYL ADD/
9592	064546	044524	043516	052040	
9593	064554	020117	051127	052111	
9594	064562	020105	044504	045523	
9595	064570	041440	046131	040440	
9596	064576	042104	000		
9597	064601	101	052124	046505	EM20: .ASCIZ /ATTEMPTING TO WRITE MAINT REG 1/
9598	064606	052120	047111	020107	
9599	064614	047524	053440	044522	
9600	064622	042524	046440	044501	
9601	064630	052116	051040	043505	
9602	064636	030440	000		
9603	064641	101	052124	046505	EM21: .ASCII /ATTEMPTING TO CLEAR BUS ADD /
9604	064646	052120	047111	020107	
9605	064654	047524	041440	042514	
9606	064662	051101	041040	051525	
9607	064670	040440	042104	040	
9608	064675	127	052111	020110	.ASCIZ /WITH A SUB CLEAR/
9609	064702	020101	052523	020102	
9610	064710	046103	040505	000122	
9611	064716	052101	042524	050115	EM22: .ASCII /ATTEMPTING TO CLEAR CS1 /
9612	064724	044524	043516	052040	
9613	064732	020117	046103	040505	
9614	064740	020122	051503	020061	
9615	064746	044527	044124	040440	.ASCIZ /WITH A SUB CLEAR/
9616	064754	051440	041125	041440	
9617	064762	042514	051101	000	
9618	064767	101	052124	046505	EM23: .ASCIZ /ATTEMPTING TO CLEAR RK611 WITH A SUBSYSTEM CLEAR/
9619	064774	052120	047111	020107	
9620	065002	047524	041440	042514	
9621	065010	051101	051040	033113	
9622	065016	030461	053440	052111	
9623	065024	020110	020101	052523	
9624	065032	051502	051531	042524	
9625	065040	020115	046103	040505	
9626	065046	000122			
9627	065050	047111	051124	042040	EM24: .ASCII /INTR DID NOT OCCUR WHEN READY AND/<15><12>
9628	065056	042111	047040	052117	
9629	065064	047440	041503	051125	
9630	065072	053440	042510	020116	
9631	065100	042522	042101	020131	



9632	065106	047101	006504	012	
9633	065113	111	052116	020122	.ASCIZ /INTR ENABLE SET IN CS1/
9634	065120	047105	041101	042514	
9635	065126	051440	052105	044440	
9636	065134	020116	051503	000061	
9637	065142	047125	054105	042520	EM25: .ASCIZ /UNEXPECTED INTR WHEN PROCESSOR PRIORITY LOWERED/
9638	065150	052103	042105	044440	
9639	065156	052116	020122	044127	
9640	065164	047105	050040	047522	
9641	065172	042503	051523	051117	
9642	065200	050040	044522	051117	
9643	065206	052111	020131	047514	
9644	065214	042527	042522	000104	
9645	065222	047111	051124	053040	EM26: .ASCIZ /INTR VECTOR ADDRESS INCORRECT/
9646	065230	041505	047524	020122	
9647	065236	042101	051104	051505	
9648	065244	020123	047111	047503	
9649	065252	051122	041505	000124	
9650	065260	047111	051124	042040	EM27: .ASCIZ /INTR DID NOT CLEAR IN RK611/
9651	065266	042111	047040	052117	
9652	065274	041440	042514	051101	
9653	065302	044440	020116	045522	
9654	065310	030465	000061		
9655	065314	054105	042520	052103	EM28: .ASCIZ /EXPECTED INTR DID NOT OCCUR AT PROCESSOR PRIORITY/
9656	065322	042105	044440	052116	
9657	065330	020122	044504	020104	
9658	065336	047516	020124	041517	
9659	065344	052503	020122	052101	
9660	065352	050040	047522	042503	
9661	065360	051523	051117	050040	
9662	065366	044522	051117	052111	
9663	065374	000131			
9664	065376	047125	054105	042520	EM29: .ASCIZ /UNEXPECTED INTR OCCURRED AT PROCESSOR PRIORITY/
9665	065404	052103	042105	044440	
9666	065412	052116	020122	041517	
9667	065420	052503	051122	042105	
9668	065426	040440	020124	051120	
9669	065434	041517	051505	047523	
9670	065442	020122	051120	047511	
9671	065450	044522	054524	000	
9672	065455	111	052116	020122	EM30: .ASCIZ /INTR DID NOT OCCUR WHEN PRIORITY LOWERED/
9673	065462	044504	020104	047516	
9674	065470	020124	041517	052503	
9675	065476	020122	044127	047105	
9676	065504	050040	044522	051117	
9677	065512	052111	020131	047514	
9678	065520	042527	042522	000104	
9679	065526	042523	052124	047111	EM31: .ASCIZ /SETTING INTR ENABLE CAUSED EXPECTED INTR/
9680	065534	020107	047111	051124	
9681	065542	042440	040516	046102	
9682	065550	020105	040503	051525	
9683	065556	042105	042440	050130	
9684	065564	041505	042524	020104	
9685	065572	047111	051124	000	
9686	065577	103	047117	051124	EM32: .ASCIZ /CONTROLLER CLEAR DID CLEAR PENDING INTR/
9687	065604	046117	042514	020122	







9744	066267	125	042516	050130	EM41: .ASCIZ /UNEXPECTED INTR DUE TO UNCLEARED CONTROLLER ERROR/
9745	066274	041505	042524	020104	
9746	066302	047111	051124	042040	
9747	066310	042525	052040	020117	
9748	066316	047125	046103	040505	
9749	066324	042522	020104	047503	
9750	066332	052116	047522	046114	
9751	066340	051105	042440	051122	
9752	066346	051117	000		
9753	066351	103	047117	051124	EM42: .ASCII /CONTROLLER CLEAR DID NOT CLEAR PENDING INTR/<15><12>
9754	066356	046117	042514	020122	
9755	066364	046103	040505	020122	
9756	066372	044504	020104	047516	
9757	066400	020124	046103	040505	
9758	066406	020122	042520	042116	
9759	066414	047111	020107	047111	
9760	066422	051124	005015		
9761	066426	052504	020105	047524	.ASCIZ /DUE TO CONTROLLER ERROR/
9762	066434	041440	047117	051124	
9763	066442	046117	042514	020122	
9764	066450	051105	047522	000122	
9765	066456	047503	052116	047522	EM43: .ASCIZ /CONTROLLER ERROR CAUSED INTR WITH INTR ENABLE RESET/
9766	066464	046114	051105	042440	
9767	066472	051122	051117	041440	
9768	066500	052501	042523	020104	
9769	066506	047111	051124	053440	
9770	066514	052111	020110	047111	
9771	066522	051124	042440	040516	
9772	066530	046102	020105	042522	
9773	066536	042523	000124		
9774	066542	047111	051124	042040	EM44: .ASCII /INTR DID NOT OCCUR WHEN INTR ENABLE SET/
9775	066550	042111	047040	052117	
9776	066556	047440	041503	051125	
9777	066564	053440	042510	020116	
9778	066572	047111	051124	042440	
9779	066600	040516	046102	020105	
9780	066606	042523	124		
9781	066611	015	053412	052111	.ASCIZ <15><12>/WITH INTR PENDING DUE TO DATA LATE/
9782	066616	020110	047111	051124	
9783	066624	050040	047105	044504	
9784	066632	043516	042040	042525	
9785	066640	052040	020117	040504	
9786	066646	040524	046040	052101	
9787	066654	000105			
9788	066656	051503	020061	047111	EM1000: .ASCIZ /CS1 INCORRECT/
9789	066664	047503	051122	041505	
9790	066672	000124			
9791	066674	047527	042122	041440	EM1001: .ASCIZ /WORD COUNT INCORRECT/
9792	066702	052517	052116	044440	
9793	066710	041516	051117	042522	
9794	066716	052103	000		
9795	066721	102	051525	040440	EM1002: .ASCIZ /BUS ADD INCORRECT/
9796	066726	042104	044440	041516	
9797	066734	051117	042522	052103	
9798	066742	000			
9799	066743	104	051511	020113	EM1003: .ASCIZ /DISK ADD INCORRECT/



9800	066750	042101	020104	047111	
9801	066756	047503	051122	041505	
9802	066764	000124			
9803	066766	051503	020062	047111	EM1004: .ASCIZ /CS2 INCORRECT/
9804	066774	047503	051122	041505	
9805	067002	000124			
9806	067004	051104	053111	020105	EM1005: .ASCIZ /DRIVE STATUS REG INCORRECT/
9807	067012	052123	052101	051525	
9808	067020	051040	043505	044440	
9809	067026	041516	051117	042522	
9810	067034	052103	000		
9811	067037	105	051122	051117	EM1006: .ASCIZ /ERROR REG INCORRECT/
9812	067044	051040	043505	044440	
9813	067052	041516	051117	042522	
9814	067060	052103	000		
9815	067063	101	052124	047105	EM1007: .ASCIZ "ATTENTION SUMMARY/OFFSET INCORRECT"
9816	067070	044524	047117	051440	
9817	067076	046525	040515	054522	
9818	067104	047457	043106	042523	
9819	067112	020124	047111	047503	
9820	067120	051122	041505	000124	
9821	067126	040504	040524	041040	EM1008: .ASCIZ /DATA BUFFER INCORRECT/
9822	067134	043125	042506	020122	
9823	067142	047111	047503	051122	
9824	067150	041505	000124		
9825	067154	040515	047111	020124	EM1009: .ASCIZ /MAINT REG 1 INCORRECT/
9826	067162	042522	020107	020061	
9827	067170	047111	047503	051122	
9828	067176	041505	000124		
9829	067202	041505	020103	047520	EM1012: .ASCIZ /ECC POS INCORRECT/
9830	067210	020123	047111	047503	
9831	067216	051122	041505	000124	
9832	067224	041505	020103	040520	EM1013: .ASCIZ /ECC PAT INCORRECT/
9833	067232	020124	047111	047503	
9834	067240	051122	041505	000124	
9835	067246	051503	020061	044103	EM1014: .ASCIZ /CS1 CHANGED AFTER READING ALL REGS/
9836	067254	047101	042507	020104	
9837	067262	043101	042524	020122	
9838	067270	042522	042101	047111	
9839	067276	020107	046101	020114	
9840	067304	042522	051507	000	
9841	067311	103	031123	041440	EM1015: .ASCIZ /CS2 CHANGED AFTER READING ALL REGS/
9842	067316	040510	043516	042105	
9843	067324	040440	052106	051105	
9844	067332	051040	040505	044504	
9845	067340	043516	04044	046114	
9846	067346	051040	043505	000123	
9847	067354	054503	020114	042101	EM1016: .ASCIZ /CYL ADD INCORRECT/
9848	067362	020104	047111	047503	
9849	067370	051122	041505	000124	
9850	067376	051503	020061	040527	EM1017: .ASCIZ /CS1 WAS MODIFIED/
9851	067404	020123	047515	044504	
9852	067412	044506	042105	000	
9853	067417	127	051117	020104	EM1018: .ASCIZ /WORD COUNT WAS MODIFIED/
9854	067424	047503	047125	020124	
9855	067432	040527	020123	047515	



9856	067440	044504	044506	042105	
9857	067446	000			
9858	067447	102	051525	040440	EM1019: .ASCIZ /BUS ADD WAS MODIFIED/
9859	067454	042104	053440	051501	
9860	067462	046440	042117	043111	
9861	067470	042511	000104		
9862	067474	044504	045523	040440	EM1020: .ASCIZ /DISK ADDRESS REG WAS MODIFIED/
9863	067502	042104	042522	051523	
9864	067510	051040	043505	053440	
9865	067516	051501	046440	042117	
9866	067524	043111	042511	000104	
9867	067532	051503	020062	040527	EM1021: .ASCIZ /CS2 WAS MODIFIED/
9868	067540	020123	047515	044504	
9869	067546	044506	042105	000	
9870	067553	104	044522	042526	EM1022: .ASCIZ /DRIVE STATUS REG WAS MODIFIED/
9871	067560	051440	040524	052524	
9872	067566	020123	042522	020107	
9873	067574	040527	020123	047515	
9874	067602	044504	044506	042105	
9875	067610	000			
9876	067611	105	051122	051117	EM1023: .ASCIZ /ERROR REG WAS MODIFIED/
9877	067616	051040	043505	053440	
9878	067624	051501	046440	042117	
9879	067632	043111	042511	000104	
9880	067640	052101	042524	052116	EM1024: .ASCIZ "ATTENTION SUMMARY/OFFSET WAS MODIFIED"
9881	067646	047511	020116	052523	
9882	067654	046515	051101	027531	
9883	067662	043117	051506	052105	
9884	067670	053440	051501	046440	
9885	067676	042117	043111	042511	
9886	067704	000104			
9887	067706	054503	020114	042101	EM1025: .ASCIZ /CYL ADD WAS MODIFIED/
9888	067714	020104	040527	020123	
9889	067722	047515	044504	044506	
9890	067730	042105	000		
9891	067733	115	044501	052116	EM1026: .ASCIZ /MAINT REG 1 WAS MODIFIED/
9892	067740	051040	043505	030440	
9893	067746	053440	051501	046440	
9894	067754	042117	043111	042511	
9895	067762	000104			
9896	067764	041505	020103	047520	EM1029: .ASCIZ /ECC POS WAS MODIFIED/
9897	067772	020123	040527	020123	
9898	070000	047515	044504	044506	
9899	070006	042105	000		
9900	070011	105	041503	050040	EM1030: .ASCIZ /ECC PAT WAS MODIFIED/
9901	070016	052101	053440	051501	
9902	070024	046440	042117	043111	
9903	070032	042511	000104		
9904					
9905	070036	000400			
9906	071036	100	100		
9907	071040	101	101		
9908	071042	102	102		
9909	071044	103	103		
9910	071046	104	104		
9911	071050	105	105		

.EVEN  
SAVVEC: .BLKW 400 ; STORAGE FOR TRAP CATCHER  
SILO: .BYTE 100,100 ; CONFIGURATION FOR SILO TEST  
.BYTE 101,101  
.BYTE 102,102  
.BYTE 103,103  
.BYTE 104,104  
.BYTE 105,105



9912	071052	106	106	.BYTE	106,106
9913	071054	107	107	.BYTE	107,107
9914	071056	110	110	.BYTE	110,110
9915	071060	111	111	.BYTE	111,111
9916	071062	112	112	.BYTE	112,112
9917	071064	113	113	.BYTE	113,113
9918	071066	114	114	.BYTE	114,114
9919	071070	115	115	.BYTE	115,115
9920	071072	116	116	.BYTE	116,116
9921	071074	117	117	.BYTE	117,117
9922	071076	120	120	.BYTE	120,120
9923	071100	121	121	.BYTE	121,121
9924	071102	122	122	.BYTE	122,122
9925	071104	123	123	.BYTE	123,123
9926	071106	124	124	.BYTE	124,124
9927	071110	125	125	.BYTE	125,125
9928	071112	126	126	.BYTE	126,126
9929	071114	127	127	.BYTE	127,127
9930	071116	130	130	.BYTE	130,130
9931	071120	131	131	.BYTE	131,131
9932	071122	132	132	.BYTE	132,132
9933	071124	133	133	.BYTE	133,133
9934	071126	134	134	.BYTE	134,134
9935	071130	135	135	.BYTE	135,135
9936	071132	136	136	.BYTE	136,136
9937	071134	137	137	.BYTE	137,137
9938	071136	140	140	.BYTE	140,140
9939	071140	141	141	.BYTE	141,141
9940	071142	142	142	.BYTE	142,142
9941	071144	143	143	.BYTE	143,143
9942	071146	144	144	.BYTE	144,144
9943	071150	145	145	.BYTE	145,145
9944	071152	146	146	.BYTE	146,146
9945	071154	147	147	.BYTE	147,147
9946	071156	150	150	.BYTE	150,150
9947	071160	151	151	.BYTE	151,151
9948	071162	152	152	.BYTE	152,152
9949	071164	153	153	.BYTE	153,153
9950	071166	154	154	.BYTE	154,154
9951	071170	155	155	.BYTE	155,155
9952	071172	156	156	.BYTE	156,156
9953	071174	157	157	.BYTE	157,157
9954	071176	160	160	.BYTE	160,160
9955	071200	161	161	.BYTE	161,161
9956	071202	162	162	.BYTE	162,162
9957	071204	163	163	.BYTE	163,163
9958	071206	164	164	.BYTE	164,164
9959	071210	165	165	.BYTE	165,165
9960	071212	166	166	.BYTE	166,166
9961	071214	167	167	.BYTE	167,167
9962	071216	170	170	.BYTE	170,170
9963	071220	171	171	.BYTE	171,171
9964	071222	172	172	.BYTE	172,172
9965	071224	173	173	.BYTE	173,173
9966	071226	174	174	.BYTE	174,174
9967	071230	175	175	.BYTE	175,175



CZR6AD0 RK611 DSKLS CTRL PRT1  
CZR6AD.P11 14-SEP-81 13:43

MACY11 30(1046) 14-SEP-81 15:04  
ERROR MESSAGES

E 15  
PAGE 187

SEQ 0186

9968	071232	176	176	.BYTE	176.176
9969	071234	177	177	.BYTE	177.177
9970	071236	200	200	.BYTE	200.200
9971	071240	201	201	.BYTE	201.201
9972	000001				

.END



ABASE = 177440	1066#	1301	1342	
ABORT = 062246	8636	9364#		
ACDW1 = 000000	1301	1344		
ACDW2 = 000000	1301	1345		
ACLO = 000010	1161#			
ACPUOP= 000000	1301	1316		
ADDW0 = 000000	1301			
ADDW1 = 000000	1301			
ADDW10= 000000	1301			
ADDW11= 000000	1301			
ADDW12= 000000	1301			
ADDW13= 000000	1301			
ADDW14= 000000	1301			
ADDW15= 000000	1301			
ADDW2 = 000000	1301			
ADDW3 = 000000	1301			
ADDW4 = 000000	1301			
ADDW5 = 000000	1301			
ADDW6 = 000000	1301			
ADDW7 = 000000	1301			
ADDW8 = 000000	1301			
ADDW9 = 000000	1301			
ADEVCT= 000000	1301	1307		
ADEVM = 000000	1301	1343		
AENV = 000000	1301	1312		
AENVM = 000000	1301	1313		
AFATAL= 000000	1301	1304		
AMADR1= 000000	1301	1329		
AMADR2= 000000	1301	1333		
AMADR3= 000000	1301	1336		
AMADR4= 000000	1301	1339		
AMAMS1= 000000	1301	1323		
AMAMS2= 000000	1301	1331		
AMAMS3= 000000	1301	1334		
AMAMS4= 000000	1301	1337		
AMSGAD= 000000	1301	1309		
AMSGLG= 000000	1301	1310		
AMSGTY= 000000	1301	1303		
AMTYP1= 000000	1301	1324		
AMTYP2= 000000	1301	1332		
AMTYP3= 000000	1301	1335		
AMTYP4= 000000	1301	1338		
APASS = 000000	1301	1306		
APRIOR= 000005	1065#	1301		
APTCSU= 000040	8513#	8675		
APTENV= 000001	8469	8511#	8545	8668
APTSIZ= 000200	1639	8510#		
APTSPO= 000100	8471	8512#	8670	
ASWREG= 000000	1301	1314		
ATESTN= 000000	1301	1305		
AUNIT = 000000	1301	1308		
AUSWR = 000000	1301	1315		
AVECT1= 120210	1064#	1301	1340	
AVECT2= 000000	1301	1341		
BAI = 000020	1124#			
BA16 = 000400	1109#			



























INTR = 000300  
IOTVEC = 000020  
IR = 000100

1102#													
1054#	1608*	1609*											
1126#	1804	1806	1852	1854	1898	1900	1946	1948	1992	1994	2113	2115	
2214	2216	2316	2339	2341	2440	2463	2465	2564	2587	2589	2688	2711	
2713	2836	2838	2968	2970	3100	3102	3232	3234	3344	3346	3420	3443	
3445	3539	3562	3564	3658	3681	3683	3777	3800	3802	3900	3923	3925	
4022	4045	4047	4144	4167	4169	4266	4289	4291	4388	4411	4413	4510	
4533	4535	4632	4655	4657	4754	4777	4779	4869	4878	4947	4994	5003	
5072	5119	5128	5197	5244	5253	5322	5387	5389	5471	5473	5556	5558	
5640	5642	5724	5726	5802	5831	5833	5922	5951	5953	6042	6071	6073	
6162	6191	6193	6287	6316	6318	6412	6441	6443	6537	6566	6568	6662	
6691	6693	6790	6792	6875	6877	7333	7347	7411	7413	7462	7475	7550	
7552	7666	7668	7782	7784	7898	7900	7975	8007	8031	8037	8080		

LF = 000012  
MCLK = 000400  
MDS = 001000  
MEMBAS = 172100  
MEMERR = 055354  
MEMVEC = 000114  
MERD = 001000  
MESMSK = 000017  
MEWD = 002000

960#	8735	8741											
1180#													
1129#	4868	4993	5118	5243									
1062#	8282												
8286	8299#												
1061#	8286*	8287*											
1181#													
1174#													
1182#	1825	1831	1919	1925	2013	2134	2235	2363	2487	2611	2735	2860	
2992	3124	3256	3365	3467	3586	3705	3824	3947	4069	4191	4313	4435	
4557	4679	4801	4924	5049	5174	5299	5408	5492	5577	5661	5745	5849	
5969	6089	6209	6274	6353	6399	6478	6524	6603	6649	6728	6811	6896	
7432	7571	7687	7803	7919									

MIND = 000200  
MSP = 000100  
NED = 010000  
NEM = 004000  
NEWPAS = 003040  
NXF = 000004  
OFFSET = 000015  
OFST = 000004  
OPI = 020000  
OPR001 = 062134  
OPR002 = 062163  
OPR003 = 062171  
OPR004 = 062221  
OR = 000200

1179#													
1178#													
1132#	4868	4993	5118	5243									
1131#	4868	4993	5118	5243									
1717	1726#	8274											
1141#													
1095#													
1160#													
1152#													
1667	9348#												
1670	1681	1698	9352#										
1677	9354#												
1689	9359#												
1127#	4868	4993	5118	5243	7411	7413	7550	7552	7666	7668	7782	7784	
7898	7900	7975	7998	8007	8068								

PACK = 000003  
PARM = 002030  
PAR.EN = 000001  
PAT = 000020  
PCA = 004000  
PCD = 010000  
PGE = 002000  
PIP = 020000  
PIRQ = 177772  
PIRQVE = 000240  
PREREG = 002014

1090#													
1212	1586#												
1063#	8285												
1176#													
1183#	6273	6398	6523	6648									
1184#	6273	6398	6523	6648									
1130#	4868	4993	5118	5243									
1168#													
966#													
1060#													
1578#	2032*	2057*	2260*	2383*	2507*	2631*	2755*	2885*	3017*	3149*	3281*	3487*	
3606*	3725*	3844*	3967*	4089*	4211*	4333*	4455*	4577*	4699*	4821*	4944*	5069*	
5194*	5319*	5356*	5365*	5869*	5989*	6109*	6229*	6350*	6475*	6600*	6725*	9236	
1580#	7192*	7209*	7210*	7211*	7212*	7213*	7214*	7221*	9241				
983#													

PRIOR = 002020  
PRO = 000000



PR1 = 000040	984#													
PR2 = 000100	985#													
PR3 = 000140	986#													
PR4 = 000200	987#													
PR5 = 000240	988#	1570												
PR6 = 000300	989#													
PR7 = 000340	990#	1595	1728	1745	6943	6958	7138	7198	7229	7262	7270	7298	7305	
	8106	8115	8130	8161	8196	8207	8281	8287	9159	9172				
PS = 177776	963#	964												
PSW = 177776	964#													
PWRVEC = 000024	1055#	1614*	1615*	9158*	9159*	9171*	9172*							
RDCHR = 104410	9001	9222#												
RDDATA = 000021	1097#													
RDGATE = 100000	1187#	6273	6398	6523	6648									
RDHEAD = 000025	1099#													
RDLIN = 104411	9073	9223#												
RDOCT = 104412	1671	1682	1699	9224#										
RDY = 000200	1108#	1783	1785	1846	1848	1877	1879	1940	1942	1976	1978	2092	2094	
	2193	2195	2314	2438	2562	2686	2805	2888	2937	3020	3069	3152	3201	
	3284	3318	3320	3418	3537	3656	3775	3898	4020	4142	4264	4386	4508	
	4630	4752	4876	5001	5126	5251	5362	5366	5445	5447	5530	5532	5614	
	5616	5698	5700	5800	5920	6040	6160	6285	6410	6535	6660	6764	6766	
	6849	6851	6956	7196	7219	7293	7334	7346	7382	7384	7461	7474	7521	
	7523	7637	7639	7753	7755	7869	7871	7974	8038	8069	8079			
RECAL = 000013	1094#													
RESREG = 104414	8630	9226#												
RESTRT = 002040	1210	1589#												
RESVEC = 000010	1050#													
RKASOF = 000016	1077#	1798	1892	1986	2107	2208	2332	2456	2580	2704	2829	2961	3093	
	3225	3338	3436	3555	3674	3793	3916	4038	4160	4282	4376*	4377	4455	
	4457	4498*	4499	4577	4579	4620*	4621	4699	4701	4742*	4743	4821	4823	
	4900	5025	5150	5275	5381	5465	5550	5634	5718	5824	5944	6064	6184	
	6309	6434	6559	6684	6784	6869	7402	7541	7657	7773	7889			
RKBA = 000004	1072#	1788	1882	1968*	1969	2036	2053*	2055	2097	2198	2304*	2305	2383	
	2385	2428*	2429	2507	2509	2552*	2553	2631	2633	2676*	2677	2755	2757	
	2811	2943	3075	3207	3323	3424	3543	3662	3781	3904	4026	4148	4270	
	4392	4514	4636	4758	4882	5007	5132	5257	5351*	5354	5450	5535	5619	
	5703	5806	5926	6046	6166	6291	6416	6541	6666	6769	6854	7387	7526	
	7642	7758	7874											
RKCS1 = 000000	1070#	1782	1845	1875*	1876	1939	1967*	1975	2052*	2054*	2083*	2091	2153*	
	2182*	2192	2254*	2298*	2313	2384*	2422*	2437	2508*	2546*	2561	2632*	2670*	
	2685	2756*	2795*	2801*	2802	2885	2886*	2887	2927*	2933*	2934	3017	3018*	
	3019	3059*	3065*	3066	3149	3150*	3151	3191*	3197*	3198	3281	3282*	3283	
	3314*	3317	3404*	3417	3488*	3523*	3536	3607*	3642*	3655	3726*	3761*	3774	
	3845*	3882*	3897	3968*	4004*	4019	4090*	4126*	4141	4212*	4248*	4263	4334*	
	4370*	4385	4456*	4492*	4507	4578*	4614*	4629	4700*	4736*	4751	4822*	4859*	
	4875	4945*	4984*	5000	5070*	5109*	5125	5195*	5234*	5250	5320*	5349*	5352*	
	5361	5441*	5444	5526*	5529	5610*	5613	5694*	5697	5784*	5799	5870*	5904*	
	5919	5990*	6024*	6039	6110*	6144*	6159	6230*	6264*	6284	6351*	6389*	6409	
	6476*	6514*	6534	6601*	6639*	6659	6726*	6760*	6763	6845*	6848	6947*	6956*	
	7196*	7219*	7259*	7268*	7292*	7293*	7294*	7295*	7329*	7332	7343*	7344	7377*	
	7381	7459	7472	7485*	7517*	7520	7633*	7636	7749*	7752	7865*	7868	7965*	
	7972	7984	7996	8015	8036	8047*	8060*	8067	8076*	8077	8103*	8112*	8137*	
	8156*	8158*	8162*	8193*	8205*	8223*								
RKCS2 = 000010	1074#	1803	1851	1897	1945	1991	2112	2213	2338	2462	2586	2710	2835	
	2967	3099	3231	3343	3442	3561	3680	3799	3922	4044	4166	4288	4410	



	4532	4654	4776	4865*	4866	4944	4946	4990*	4991	5069	5071	5115*	5116
	5194	5196	5240*	5241	5319	5321	5353*	5386	5470	5555	5639	5723	5830
	5950	6070	6190	6315	6440	6565	6690	6789	6874	7331	7345	7410	7460
	7473	7549	7665	7781	7897	7973	7985	7997	8016	8035	8065	8078	
RKDA = 000006	1073#	1793	1887	1981	2102	2184*	2186	2255	2326	2450	2574	2698	2823
	2955	3087	3219	3333	3430	3549	3668	3787	3888*	3889	3967	3969	4010*
	4011	4089	4091	4132*	4133	4211	4213	4254*	4255	4333	4335	4404	4526
	4648	4770	4894	5019	5144	5269	5376	5460	5545	5629	5713	5818	5938
	6058	6178	6303	6428	6553	6678	6779	6864	7397	7536	7652	7768	7884
RKDB = 000024	1079#	7330	7380*	7451	7471	7519*	7590	7635*	7706	7751*	7822	7867*	7938
	7968*	7983*	7995*	8008	8014	8034	8062*	8113	8157	8202			
RKDCYL= 000020	1078#	1819	1913	2007	2128	2229	2356	2480	2604	2728	2853	2985	3117
	3249	3359	3460	3579	3698	3817	3940	4062	4184	4306	4428	4550	4672
	4794	4917	5042	5167	5292	5402	5486	5571	5655	5739	5790*	5791	5869
	5871	5910*	5911	5989	5991	6030*	6031	6109	6111	6150*	6151	6229	6231
	6333	6458	6583	6708	6805	6890	7426	7565	7681	7797	7913		
RKDS = 000012	1075#	1809	1903	1997	2118	2219	2345	2469	2593	2717	2842	2974	3106
	3238	3349	3449	3568	3687	3806	3929	4051	4173	4295	4417	4539	4661
	4783	4906	5031	5156	5281	5392	5443*	5476	5561	5645	5729	5837	5957
	6077	6197	6322	6447	6572	6697	6795	6880	7416	7555	7671	7787	7903
RKECPS= 000030	1083#	1839	1933	2026	2147	2248	2377	2501	2625	2749	2874	3006	3138
	3270	3378	3481	3600	3719	3838	3961	4083	4205	4327	4449	4571	4693
	4815	4938	5063	5188	5313	5421	5505	5590	5674	5758	5863	5983	6103
	6223	6344	6469	6594	6719	6824	6847*	6909	7445	7584	7700	7816	7932
RKECPT= 000032	1084#	1834	1928	2021	2142	2243	2372	2496	2620	2744	2869	3001	3133
	3265	3373	3476	3595	3714	3833	3956	4078	4200	4322	4444	4566	4688
	4810	4933	5058	5183	5308	5416	5500	5585	5669	5753	5858	5978	6098
	6218	6339	6464	6589	6714	6762*	6819	6904	7440	7579	7695	7811	7927
RKER = 000014	1076#	1814	1908	2002	2123	2224	2350	2474	2598	2722	2847	2979	3111
	3243	3354	3454	3573	3692	3811	3934	4056	4178	4300	4422	4544	4666
	4788	4911	5036	5161	5286	5397	5481	5528*	5566	5650	5734	5842	5962
	6082	6202	6327	6452	6577	6702	6800	6885	7421	7560	7676	7792	7908
RKMR1 = 000026	1080#	1824	1918	2012	2133	2234	2362	2486	2610	2734	2859	2991	3123
	3255	3364	3466	3585	3704	3823	3946	4068	4190	4312	4434	4556	4678
	4800	4923	5048	5173	5298	5407	5491	5576	5660	5744	5848	5968	6088
	6208	6270*	6271	6350	6352	6395*	6396	6475	6477	6520*	6521	6600	6602
	6645*	6646	6725	6727	6810	6895	7431	7570	7686	7802	7918		
RKMR2 = 000034	1081#	5612*											
RKMR3 = 000036	1082#	5696*											
RKPRI 002002	1570#	1715*	7184	7209	7215								
RKSPAR= 000022	1085#	3316*											
RKVEC 002000	1569#	1713*	1714*	7120	7182	7207*	7222*	7244	7260	7276	7296	7311	8104
	8123*	8138	8159	8175	8194	8204*	8224	9239					
RKWC = 000002	1071#	2084*	2085	2154	2183*	2203	2303*	2320	2427*	2444	2551*	2568	2675*
	2692	2800*	2817	2932*	2949	3064*	3081	3196*	3213	3315*	3328	3409*	3410
	3487	3489	3528*	3529	3606	3608	3647*	3648	3725	3727	3766*	3767	3844
	3846	3887*	3910	4009*	4032	4131*	4154	4253*	4276	4375*	4398	4497*	4520
	4619*	4642	4741*	4764	4864*	4888	4989*	5013	5114*	5138	5239*	5263	5350*
	5371	5442*	5455	5527*	5540	5611*	5624	5695*	5708	5789*	5812	5909*	5932
	6029*	6052	6149*	6172	6269*	6297	6394*	6422	6519*	6547	6644*	6672	6761*
	6774	6846*	6859	7378*	7392	7518*	7531	7634*	7647	7750*	7763	7866*	7879
RLS = 000010	1123#												
SAVFLG 002016	1579#	1716	1724*	6938*	6968*	7112*	7128*	7150*	7163*				
SAVREG= 104413	8579	9225#											
SAVSWR 002026	1583#	9157*	9176										
SAVVEC 070036	1718	6932	6962	7106	7122	7144	7157	9905#					







S.UNLD=	002000	1197#		
TBITVE=	000014	1051#		
TKVEC =	000060	1058#		
TPVEC =	000064	1059#		
TRAPPC	002024	1582#	8300*	9230
TRAPVE=	000034	1057#	1612*	1613*
TRTVEC=	000014	1052#		
TST1	003062	1729	1741#	8385
TST10	006702	2256	2293#	8392
TST11	007530	2396	2417#	8393
TST12	010356	2520	2541#	8394
TST13	011202	2644	2665#	8395
TST14	012026	2768	2789#	8396
TST15	012714	2900	2921#	8397
TST16	013604	3032	3053#	8398
TST17	014470	3164	3185#	8399
TST2	003154	1776#	8386	
TST20	015354	3296	3309#	8400
TST21	016062	3380	3399#	8401
TST22	016700	3501	3518#	8402
TST23	017516	3620	3637#	8403
TST24	020332	3739	3756#	8404
TST25	021146	3858	3877#	8405
TST26	021774	3980	3999#	8406
TST27	022622	4102	4121#	8407
TST3	003714	1853	1871#	8387
TST30	023446	4224	4243#	8408
TST31	024272	4346	4365#	8409
TST32	025120	4468	4487#	8410
TST33	025746	4590	4609#	8411
TST34	026572	4712	4731#	8412
TST35	027416	4834	4853#	8413
TST36	030266	4959	4978#	8414
TST37	031136	5084	5103#	8415
TST4	004454	1947	1962#	8388
TST40	032002	5209	5228#	8416
TST41	032646	5334	5346#	8417
TST42	033412	5423	5436#	8418
TST43	034120	5507	5521#	8419
TST44	034626	5592	5605#	8420
TST45	035334	5676	5689#	8421
TST46	036042	5760	5779#	8422
TST47	036670	5882	5899#	8423
TST5	005210	2037	2049#	8389
TST50	037516	6002	6019#	8424
TST51	040342	6122	6139#	8425
TST52	041166	6242	6259#	8426
TST53	042032	6367	6384#	8427
TST54	042676	6492	6509#	8428
TST55	043540	6617	6634#	8429
TST56	044402	6742	6755#	8430
TST57	045110	6826	6840#	8431
TST6	005306	2056	2074#	8390
TST60	045616	6911	6930#	8432
TST61	046672	7114	7130	7151
TST62	047156	7257#	8434	

7180# 8433











5585*	5590*	5591	5613*	5614	5619*	5624*	5629*	5634*	5639*	5640	5645*	5650*	
5655*	5660*	5662	5665	5669*	5674*	5675	5697*	5698	5703*	5708*	5713*	5718*	
5723*	5724	5729*	5734*	5739*	5744*	5746	5749	5753*	5758*	5759	5791*	5794	
5799*	5800	5806*	5812*	5818*	5824*	5830*	5831	5837*	5842*	5848*	5850	5853	
5858*	5863*	5865	5871*	5873	5911*	5914	5919*	5920	5926*	5932*	5938*	5944*	
5950*	5951	5957*	5962*	5968*	5970	5973	5978*	5983*	5985	5991*	5993	6031*	
6034	6039*	6040	6046*	6052*	6058*	6064*	6070*	6071	6077*	6082*	6088*	6090	
6093	6098*	6103*	6105	6111*	6113	6151*	6154	6159*	6160	6166*	6172*	6178*	
6184*	6190*	6191	6197*	6202*	6208*	6210	6213	6218*	6223*	6225	6231*	6233	
6271*	6275	6279	6284*	6285	6291*	6297*	6303*	6309*	6315*	6316	6322*	6327*	
6333*	6339*	6344*	6346	6352*	6354	6358	6396*	6400	6404	6409*	6410	6416*	
6422*	6428*	6434*	6440*	6441	6447*	6452*	6458*	6464*	6469*	6471	6477*	6479	
6483	6521*	6525	6529	6534*	6535	6541*	6547*	6553*	6559*	6565*	6566	6572*	
6577*	6583*	6589*	6594*	6596	6602*	6604	6608	6646*	6650	6654	6659*	6660	
6666*	6672*	6678*	6684*	6690*	6691	6697*	6702*	6708*	6714*	6719*	6721	6727*	
6729	6733	6763*	6764	6769*	6774*	6779*	6784*	6789*	6790	6795*	6800*	6805*	
6810*	6812	6815	6819*	6824*	6825	6848*	6849	6854*	6859*	6864*	6869*	6874*	
6875	6880*	6885*	6890*	6895*	6897	6900	6904*	6909*	6910	7381*	7382	7387*	
7392*	7397*	7402*	7410*	7411	7416*	7421*	7426*	7431*	7433	7436	7440*	7445*	
7446	7451*	7452	7520*	7521	7526*	7531*	7536*	7541*	7549*	7550	7555*	7560*	
7565*	7570*	7572	7575	7579*	7584*	7585	7590*	7591	7636*	7637	7642*	7647*	
7652*	7657*	7665*	7666	7671*	7676*	7681*	7686*	7688	7691	7695*	7700*	7701	
7706*	7707	7752*	7753	7758*	7763*	7768*	7773*	7781*	7782	7787*	7792*	7797*	
7802*	7804	7807	7811*	7816*	7817	7822*	7823	7868*	7869	7874*	7879*	7884*	
7889*	7897*	7898	7903*	7908*	7913*	7918*	7920	7923	7927*	7932*	7933	7938*	
7939	8008*	8014*	8024	9232	9234	9236	9247						
1293#	8535	8568											
\$BELL 001204													
\$CDW1 001274													
\$CDW2 001276													
\$CHARC 057430	8690*	8700*	8707	8733*	8738#								
\$CKSWR 060100	8895#	9221											
\$CMTAG 001100	1253#	1601	1602	1610	1616	1617	1618						
\$CM3 = 000000	1285#												
\$CM4 = 000010	1283#	1284#	1285#	1286#	1287#	1288#	1289#	1290#	1291#				
\$CNTLG 060771	8906	9049#											
\$CNTLU 060764	8923	9023	9048#										
\$CPJOP 001242	1316#												
\$CRLF 001211	1295#	8263	8543	8568	8589	8594	8602	8615	8628	8689	8741	8934	9028
	9048	9107											
\$DBLK 060076	8842	8876	8884#										
\$DEVCT 001224	1307#												
\$DEVN 001272	1343#												
\$DOAGN 055254	8245	8266	8272#										
\$DTBL 060066	8845	8880#											
\$ENDAD 055244	1219	1648	8268#	8563									
\$ENDCT 055110	1616	8247#											
\$ENULL 055260	8275#												
\$ENV 001234	1312#	1654	8469	8493	8545	8668							
\$ENVN 001235	1313#	1639	8471	8670	8675								
\$EOP 055054	8237#	8642											
\$EOPCT 055102	1616*	8244#	8248	8640*									
\$ERFLG 001103	1256#	8315	8354	8356	8362*	8384	8384	8453	8530*	8568			
\$ERMAX 001115	1262#	1619*	8356	8379*	8384								
\$ERROR 056416	1610	8528#											
\$ERRPC 001116	1263#	8537*	8538*	8539	8568	9231	9232	9234	9236	9238	9239	9241	9242
	9244	9247											

\$BELL 001204  
\$CDW1 001274  
\$CDW2 001276  
\$CHARC 057430  
\$CKSWR 060100  
\$CMTAG 001100  
\$CM3 = 000000  
\$CM4 = 000010  
\$CNTLG 060771  
\$CNTLU 060764  
\$CPJOP 001242  
\$CRLF 001211  
\$DBLK 060076  
\$DEVCT 001224  
\$DEVN 001272  
\$DOAGN 055254  
\$DTBL 060066  
\$ENDAD 055244  
\$ENDCT 055110  
\$ENULL 055260  
\$ENV 001234  
\$ENVN 001235  
\$EOP 055054  
\$EOPCT 055102  
\$ERFLG 001103  
\$ERMAX 001115  
\$ERROR 056416  
\$ERRPC 001116











\$MTYP3 001255  
\$MTYP4 001261  
\$MXCNT 055726  
\$NULL 001154  
\$NWTST= 000001

1335#														
1338#														
8373#	8383#													
1279#	8695	8741												
1731#	1733	1762#	1764	1858#	1860	1952#	1954	2042#	2044	2063#	2065	2168#		
2170	2275#	2277	2399#	2401	2523#	2525	2647#	2649	2771#	2773	2903#	2905		
3035#	3037	3167#	3169	3299#	3301	3385#	3387	3504#	3506	3623#	3625	3742#		
3744	3861#	3863	3983#	3985	4105#	4107	4227#	4229	4349#	4351	4471#	4473		
4593#	4595	4715#	4717	4837#	4839	4962#	4964	5087#	5089	5212#	5214	5337#		
5339	5427#	5429	5512#	5514	5597#	5599	5681#	5683	5765#	5767	5885#	5887		
6005#	6007	6125#	6127	6245#	6247	6370#	6372	6495#	6497	6620#	6622	6745#		
6747	6831#	6833	6919#	6921	7168#	7170	7249#	7251	7281#	7283	7319#	7321		
7357#	7359	7487#	7489	7603#	7605	7719#	7721	7835#	7837	7951#	7953	8049#		
8051	8090#	8092	8143#	8145	8180#	8182								

\$OCNT 057656  
\$OMODE 057660  
\$OVER 055712  
\$PASS 001222  
\$PASTM 001006  
\$POWER 061376  
\$PWRCT 061372  
\$PWRDN 061254  
\$PWRUP 061302  
\$QUES 001210  
\$RDCHR 060370  
\$RDDEC= \*\*\*\*\* U  
\$RDLIN 060520  
\$RDOCT 061020  
\$RDSZ = 000010  
\$RESRE 061216  
\$RTNAD 055256  
\$R2A = \*\*\*\*\* U  
\$SAVRE 061160  
\$SCOPE 055416  
\$SETUP= 000137

8773*	8802*	8815#												
8768*	8772*	8777	8780*	8791*	8817#									
8327	8352	8361	8371	8380#										
1306#	1638*	8241*	8242*	8253	8275	8367	8384							
1243#														
9174	9180#													
9165*	9166*	9167*	9169*	9179#										
1614	9157#	9171												
9158	9165#													
1294#	8568	8741	8953	9031	9048	9104	9107							
8966#	9222													
9225														
8996#	9223													
9068#	9224													
8989#														
9140#	9226													
8274#														
9227														
9124#	9225													
1608	8324#													
1586#	1607	1608	1610	1612	1614	1616	1617	1618	1620	1648	1651	8239		
8325	8529	8555	8563	8890	9054									
1586#														

\$STUP = 177777  
\$SVLAD 055656  
\$SVPC = 000220  
\$SWR = 167400

8335	8374#													
1217#	1222													
929#	939	943	944	945	946	947	948	949	950	1291	1292	1293		
1617	1618	1620	1621	1742	1777	1872	1963	2050	2075	2178	2294	2418		
2542	2666	2790	2922	3054	3186	3310	3400	3519	3638	3757	3878	4000		
4122	4244	4366	4488	4610	4732	4854	4979	5104	5229	5347	5437	5522		
5606	5690	5780	5900	6020	6140	6260	6385	6510	6635	6756	6841	6931		
7181	7258	7291	7327	7376	7509	7625	7741	7857	7963	8058	8101	8154		
8191	8234	8240	8267	8273	8275	8316	8317	8318	8319	8320	8326	8338		
8340	8341	8354	8355	8356	8363	8364	8365	8377	8380	8383	8520	8521		
8522	8523	8524	8533	8540	8552	8556	8568							

\$SWREG 001236  
\$SWRMK= 000000  
\$SWOBT 055730  
\$TESTN 001220  
\$TIMES 001200

1314#	1641													
950	951	8320	8321	8344										
8350	8384#													
1305#	8375*	9230	9231	9232	9234	9236	9238	9239	9241	9242	9244	9247		
1291#	1617*	1742*	1777*	1872*	1963*	2050*	2075*	2178*	2294*	2418*	2542*	2666*		
2790*	2922*	3054*	3186*	3310*	3400*	3519*	3638*	3757*	3878*	4000*	4122*	4244*		
4366*	4488*	4610*	4732*	4854*	4979*	5104*	5229*	5347*	5437*	5522*	5606*	5690*		
5780*	5900*	6020*	6140*	6260*	6385*	6510*	6635*	6756*	6841*	6931*	7181*	7258*		















MSG	1731#	1733	1762#	1764	1858#	1860	1952#	1954	2042#	2044	2063#	2065	2168#	2170	2275#
	2277	2399#	2401	2523#	2525	2647#	2649	2771#	2773	2903#	2905	3035#	3037	3167#	3169
	3299#	3301	3385#	3387	3504#	3506	3623#	3625	3742#	3744	3861#	3863	3983#	3985	4105#
	4107	4227#	4229	4349#	4351	4471#	4473	4593#	4595	4715#	4717	4837#	4839	4962#	4964
	5087#	5089	5212#	5214	5337#	5339	5427#	5429	5512#	5514	5597#	5599	5681#	5683	5765#
	5767	5885#	5887	6005#	6007	6125#	6127	6245#	6247	6370#	6372	6495#	6497	6620#	6622
	6745#	6747	6831#	6833	6919#	6921	7168#	7170	7249#	7251	7281#	7283	7319#	7321	7357#
	7359	7487#	7489	7603#	7605	7719#	7721	7835#	7837	7951#	7953	8049#	8051	8090#	8092
	8143#	8145	8180#	8182											
MULT	1#	1061#													
NEWST	1#	1061#	1731	1762	1858	1952	2042	2063	2168	2275	2399	2523	2647	2771	2903
	3035	3167	3299	3385	3504	3623	3742	3861	3983	4105	4227	4349	4471	4593	4715
	4837	4962	5087	5212	5337	5427	5512	5597	5681	5765	5885	6005	6125	6245	6370
	6495	6620	6745	6831	6919	7168	7249	7281	7319	7357	7487	7603	7719	7835	7951
	8049	8090	8143	8180											
POP	1#	1061#	8503	8504	8871	9096	9145								
PUSH	1#	1061#	8464	8466	8487	8830	9070	9125							
REPORT	1#	1061#													
SCOPE	956#	1741	1776	1871	1962	2049	2074	2177	2293	2417	2541	2665	2789	2921	3053
	3185	3309	3399	3518	3637	3756	3877	3999	4121	4243	4365	4487	4609	4731	4853
	4978	5103	5228	5346	5436	5521	5605	5689	5779	5899	6019	6139	6259	6384	6509
	6634	6755	6840	6930	7180	7257	7290	7326	7375	7508	7624	7740	7856	7962	8057
	8100	8153	8190	8238											
SETPRI	1#	1061#													
SETTRA	9205#	9214	9215	9216	9217	9219	9221	9222	9223	9224	9225	9226	9227		
SETUP	1#	1061#	1599												
SKIP	1#	1061#	1853	1947	2037	2056	2164	2256	2396	2520	2644	2768	2900	3032	3164
	3296	3380	3501	3620	3739	3858	3980	4102	4224	4346	4468	4590	4712	4834	4959
	5084	5209	5334	5423	5507	5592	5676	5760	5882	6002	6122	6242	6367	6492	6617
	6742	6826	6911	7114	7130	7151	7353	7600	7716	7832	7948	8086			
SLASH	1#	1061#													
SPACE	1061#														
STARS	1#	1061#	1215	1226	1228	1235	1248	1297	1300	1731	1740	1762	1775	1858	1870
	1952	1961	2042	2048	2063	2073	2168	2176	2264	2273	2275	2292	2399	2416	2523
	2540	2647	2664	2771	2788	2903	2920	3035	3052	3167	3184	3299	3308	3385	3398
	3504	3517	3623	3636	3742	3755	3861	3876	3983	3998	4105	4120	4227	4242	4349
	4364	4471	4486	4593	4608	4715	4730	4837	4852	4962	4977	5087	5102	5212	5227
	5337	5345	5427	5435	5512	5520	5597	5604	5681	5688	5765	5778	5885	5898	6005
	6018	6125	6138	6245	6258	6370	6383	6495	6508	6620	6633	6745	6754	6831	6839
	6919	6929	7168	7179	7249	7256	7281	7289	7319	7325	7357	7374	7487	7507	7603
	7623	7719	7739	7835	7855	7951	7961	8049	8056	8090	8099	8143	8152	8180	8189
	8230	8312	8447	8459	8516	8569	8578	8647	8743	8820	8887	8890	8958	8989	9056
	9109	9155	9163	9184											
SWRSU	1#	1061#	1622#												
TRMTRP	9205#														
TYPBIN	1#	1061#													
TYPDEC	1#	1061#	8253	8260											
TYPNAM	1#	1061#	1644												
TYPNUM	1#	1061#													
TYPOCS	1#	1061#													
TYPOCT	1#	1061#	1668	8908											
TYPTXT	1#	1061#	8249	8256											
WRTDB	1246#	7510	7626	7742	7858										
WRTRG1	1246#	3311	5438	5523	5607	5691	6757	6842							
WRTRG2	1246#	2295	2419	2543	2667	2791	2923	3055	3187	3401	3520	3639	3758	3879	4001
	4123	4245	4367	4489	4611	4733	4855	4980	5105	5230	5781	5901	6021	6141	6261







CZR6ADO RK611 DSKLS CTRL PRT1 MACY11 30(1046) 14-SEP-81 15:04 E 1 PAGE 213  
CZR6AD.P11 14-SEP-81 13:43 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0210

. ABS. 071242 000

ERRORS DETECTED: 0

CZR6AD,CZR6AD.LST/SOL/CRF/NL:TOC=SYSMAC.SML,CZR6AD.P11  
RUN-TIME: 38 44 3 SECONDS  
RUN-TIME RATIO: 152/86=1.7  
CORE USED: 42K (84 PAGES)