

RM05/3/2

RM05/3/2 DU POR TST1 AH-F937A-MC  
CZRMRAO FICHE 1 OF 2

JUN 1980  
COPYRIGHT © 1980  
MADE IN USA



The main body of the document is a large grid of approximately 15 columns and 15 rows of data. Each cell in the grid contains a small, dense table of information, likely representing a detailed technical or operational record. The text within these cells is extremely small and difficult to read, but the overall structure is a regular, repeating pattern of data points.



RM05/3/2

RM05/3/2 DU POR TST1 AH-F937A-MC  
CZRMRA0 FICHE 2 OF 2

JUN 1980  
COPYRIGHT © 1980  
MADE IN USA



[Faint, illegible text visible through the paper from the reverse side, appearing as bleed-through.]





1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51

.REM    ^

IDENTIFICATION

PRODUCT CODE:    AC-F936A-MC  
PRODUCT NAME:    CZRMRAO RM05/3/2 DU POR TST 1  
DATE CREATED:    APRIL 1980  
MAINTAINER:      CX DIAGNOSTIC GROUP  
AUTHOR:          MIKE LEAVITT

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

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

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

COPYRIGHT (C) 1980 DIGITAL EQUIPMENT CORPORATION



CONTENTS  
-----

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38

- 1. ABSTRACT
- 2. REQUIREMENTS
  - 2.1 EQUIPMENT
  - 2.2 PREREQUISITE PROGRAMS
  - 2.3 OTHER PROGRAMS
- 3. LOADING PROCEDURES
- 4. STARTING PROCEDURES
  - 4.1 STARTING ADDRESSES
  - 4.2 UNIBUS & VECTOR ADDRESSES
  - 4.3 OPERATOR ACTION
- 5. OPERATING PROCEDURES
  - 5.1 OPERATIONAL SWITCH SETTINGS
  - 5.2 'SOFTWARE' SWITCH REGISTER
  - 5.3 TEST SELECTION
  - 5.4 DUAL PORT TEST CABLE CONNECTION
- 6. ERRORS
- 7. MISCELLANEOUS
  - 7.1 RESTRICTIONS
  - 7.2 LIMITATIONS
  - 7.3 EXECUTION TIME
  - 7.4 REQUIRED TESTS
  - 7.5 DISK SURFACE USAGE
  - 7.6 LOOP ON ERROR OPTION
- 8. TEST DESCRIPTIONS



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57

1.      ABSTRACT

THE RM05/3/2 DUAL PORT LOGIC TEST PERFORMS A SERIES OF TESTS WHICH VERIFY THAT THE RM05/3/2 DUAL PORT LOGIC IS FUNCTIONING PROPERLY. ONLY THE CONTROL LOGIC IS TESTED BY THIS PROGRAM; DATA HANDLING IN THE DUAL PORT MODE IS NOT TESTED BY THIS PROGRAM.

BOTH PORTS OF THE DRIVE ARE CABLED TO THE SAME MASSBUS BY A SPECIAL ADAPTER CABLE. THIS ARRANGEMENT ALLOWS THE DUAL PORT LOGIC TO BE TESTED FROM ONE PDP-11, RH11 OR RH70.

THIS PROGRAM IS THE FIRST PART OF THE DUAL PORT OPTION LOGIC TEST. THE SECOND PART OF THE TEST PERFORMS MANUAL INTERVENTION TESTS.

2.      REQUIREMENTS

2.1     EQUIPMENT

- PDP-11 PROCESSOR
- 16K OF MEMORY
- KW11-L OR KW11-P CLOCK
- TERMINAL
- RH11 OR RH70
- 1 - DISK DRIVE (RM05, RM03 OR RM02)
- RM DUAL PORT TEST CABLE

2.2     PREREQUISITE PROGRAMS

RM05/3/2 DISKLESS DIAGNOSTIC, PART 1 & 2

RM05/3/2 FUNCTIONAL TEST, PART 1, 2 & 3

THE PRELIMINARY PROGRAMS MUST BE RUN TWICE: ONCE FROM EACH CONTROLLER (PORT).

2.3     OTHER PROGRAMS

- A. THE OPERATION OF THE "PORT SELECT" SWITCH IS TESTED BY THE SECOND PART OF THE DUAL PORT LOGIC TEST.
- B. DYNAMIC OPERATION OF THE DUAL PORT OPTION IS TESTED BY THE RM05/3/2 PERFORMANCE EXERCISER PROGRAM.

3.      LOADING PROCEDURES

THE PROGRAM MAY BE LOADED BY THE ABSOLUTE PAPER TAPE LOADER OR IT MAY BE LOADED FROM THE APPROPRIATE MEDIA USING THE ASSOCIATED 'XXDP' LOADER. THE PROGRAM MAY NOT



58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114

BE INCLUDED IN AN 'XXDP' CHAIN.

4. STARTING PROCEDURES  
-----

4.1 STARTING ADDRESSES

- A. THE NORMAL STARTING ADDRESS OF THE PROGRAM IS LOCATION 200 (8). STARTING AT THIS ADDRESS ALLOWS THE OPERATOR TO SELECT (OR RESELECT) THE ADDRESS OF THE DRIVE TO BE TESTED.
- B. THE RESTART ADDRESS IS LOCATION 204 (8). THE PROGRAM WILL USE THE CURRENT DRIVE (DCL) ADDRESS.
- C. THE PROGRAM CAN BE STARTED AT LOCATION 210 (8) TO ALLOW THE ADDRESS OF THE RH11 OR RH70 TO BE CHANGED.

4.2 UNIBUS & VECTOR ADDRESSES

THE PROGRAM ASSUMES THE FOLLOWING UNIBUS AND VECTOR ADDRESSES. THESE ADDRESSES MAY BE CHANGED PRIOR TO STARTING THE PROGRAM FROM ANY OF THE STARTING ADDRESSES.

MEMORY LOCATION	CONTENTS	FUNCTION
-----	-----	-----
1142	177560	TTY KEYBOARD STATUS REG
1144	177562	TTY KEYBOARD BUFFER REG
1146	177564	TTY PRINTER STATUS REG
1150	177566	TTY PRINTER BUFFER REG
1210	172540	KW11-P STATUS REG
1212	172542	KW11-P COUNTER BUFFER
1214	104	KW11-P VECTOR ADDRESS
1216	177546	KW11-L STATUS REGISTER
1220	100	KW11-L VECTOR ADDRESS

4.3 OPERATOR ACTION

- A. CONNECT THE DUAL PORT TEST CABLE BETWEEN BUS A & BUS B ON THE DRIVE BEING TESTED. (SEE SECTION 5.4)
- B. LOAD THE PROGRAM INTO MEMORY IN THE PROCESSOR CONTROLLING THE MASSBUS USED FOR TESTING.
- C. SWITCH THE 'PORT SELECT' SWITCH ON THE DRIVE TO BE TESTED TO THE 'A/B' POSITION. CYCLE THE DRIVE UP.
- D. LOAD THE APPROPRIATE STARTING ADDRESS (200(8) OR 210(8)) INTO THE SWITCH REGISTER (OR THE 'SOFTWARE' SWITCH REGISTER. REFER TO SECTION 5.2).
- E. PRESS START.
- F. ENTER THE DRIVE NUMBER.
- G. ENTER THE NUMBER OF THE TEST TO BE RUN. ('CARRIAGE RETURN' OR '0' WILL RUN ALL TESTS.)
- H. THE PROGRAM MAY BE STOPPED AT ANY TIME AND RESTARTED FROM LOCATION 204.



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  
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

5.      OPERATING PROCEDURES  
-----

5.1      OPERATIONAL SWITCH SETTINGS

WITH ALL SWITCHES SET TO ZERO, THE PROGRAM WILL TYPE ALL ERRORS AND CONTINUE TESTING.

THE SWITCH SETTINGS ARE:

- SW<15>=1...HALT ON ERROR
- SW<14>=1...LOOP ON TEST
- SW<13>=1...INHIBIT ERROR TYPEOUTS
- SW<11>=1...INHIBIT TEST ITERATIONS
- SW<10>=1...RING TTY BELL ON ERROR
- SW<09>=1...LOOP ON ERROR

5.2      'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RM05/3/2 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN      NEW ='

EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED., 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

5.3      TEST SELECTION

INDIVIDUAL TESTS ARE SELECTED IN RESPONSE TO THE 'ENTER TEST NUMBER:' MESSAGE. ANY VALID TEST NUMBER CAN BE ENTERED. EACH ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN (CR). THE LOOP ON TEST SWITCH, SW<14>, MUST BE SET TO ALLOW CONTINUOUS EXECUTION OF THE SELECTED TEST.

TO RUN ALL TESTS IN SEQUENCE, ENTER EITHER A '0' FOLLOWED BY A CARRIAGE RETURN OR A CARRIAGE RETURN BY ITSELF. THE PROGRAM WILL THEN EXECUTE ALL TESTS IN SEQUENCE.



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  
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

THE 'RUBOUT KEY' (RO) CAN BE USED TO DELETE THE LAST CHARACTER ENTERED. SUCCESSIVELY STRIKING THE RO KEY WILL DELETE CHARACTERS UNTIL THE PREVIOUS CHARACTERS HAVE BEEN DELETED. CHARACTERS DELETED BY THE RO KEY WILL BE TYPED AND WILL BE SEPARATED BY '\ ' FROM THE CHARACTERS ENTERED BY THE OPERATOR.

THE OPERATOR CAN DELETE AN ENTIRE ENTRY BY TYPING A 'CONTROL U' .

5.4 TEST CABLE CONNECTION

TO TEST THE RM05/3/2 DUAL PORT OPTION WITH THIS PROGRAM, A SPECIAL TEST CABLE MUST BE USED. (THE TEST CABLE IS P/N 7010507-02). THE TEST CABLE CONNECTS MASSBUS A & MASSBUS B TOGETHER AT THE DRIVE BEING TESTED AND IS CONSTRUCTED SO THAT BIT 0 OF THE MASSBUS UNIT SELECT LINES IS COMPLEMENTED.

WITH THE DRIVE CABLE CONNECTED TO THE RM05/3/2 UNDER TEST, THE DRIVE APPEARS AS TWO UNITS ON THE MASSBUS: EACH PORT OF THE DRIVE WILL RESPOND TO A DIFFERENT MASSBUS ADDRESS. THE ADDRESS OF EACH PORT WILL DEPEND UPON THE DRIVE'S ADDRESS PLUG.

THE PROGRAM WILL TYPEOUT THE APPARENT ADDRESSES OF BOTH PORTS. (ONE PORT WILL HAVE THE ADDRESS OF THE DRIVE; THE OTHER PORT WILL HAVE THE ADDRESS DEVELOPED BY THE CABLE).

\*\*\*\*\*  
\* ANY OTHER DRIVE ON THE MASSBUS WHICH HAS AN ADDRESS \*  
\* IN CONFLICT WITH EITHER OF THE TEST ADDRESSES MUST BE \*  
\* POWERED DOWN. \*  
\*\*\*\*\*

THE TEST CABLE CONNECTION TO THE DRIVE UNDER TEST WILL DEPEND ON WHICH PROCESSOR, RH11/RH70 IS TO TEST THE DRIVE. IF THE DRIVE IS TO BE TESTED BY THE PROCESSOR ON PORT A, CONNECT THE MASSBUS CABLE FROM THE RH TO J3 OF THE RM05/3/2 BACK PANEL, THEN CONNECT THE TEST CABLE (P/N7010507-02) FROM J2 TO J7 OF THE BACK PANEL AND TERMINATE THE PORT 'B' AT J8.

WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION BITS FOR PORTS A & B ARE ASSERTED IN THE SAME BIT POSITION WHEN 'RMAS' (ATTENTION SUMMARY REGISTER) IS READ. THE ATTENTION BIT POSITION IS DETERMINED BY THE ADDRESS OF THE DRIVE THE ATTENTION BIT THAT APPEARS FOR THE DRIVE IS THE INCLUSIVE 'OR' OF THE PORT A & PORT B ATTENTION BITS. BECAUSE OF THIS, THE PROGRAM LOOKS AT ONLY THE ATTENTION BIT IN 'RMDS' (DRIVE STATUS REGISTER) TO DETERMINE THE STATE OF THE SELECTED PORTS'S ATTENTION BIT.

6. ERRORS  
-----



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  
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

WHEN THE PROGRAM ENCOUNTERS AN ERROR, THE ERROR ROUTINE IS CALLED AND IF SW<13> IS NOT SET, THE ERROR MESSAGE PERTAINING TO THE ERROR WILL BE TYPED. EACH ERROR TYPEOUT WILL CONTAIN THE FOLLOWING:

- A. AN ERROR MESSAGE
- B. A DATA HEADER LINE
- C. A DATA LINE CONTAINING:
  - 1. THE TEST NUMBER
  - 2. THE PC (PROGRAM COUNTER VALUE) WHERE THE ERROR CALL WAS MADE
  - 3. CONTENTS OF THE APPROPRIATE REGISTERS

7. MISCELLANEOUS  
-----

7.1 RESTRICTIONS

TO RUN THIS PROGRAM, THE SYSTEM MUST HAVE EITHER A KW11-P OR A KW11-L CLOCK. ADDITIONALLY, THE DRIVE UNDER TEST MUST HAVE THE DUAL PORT TEST CABLE CONNECTED.

7.2 LIMITATIONS

THIS PROGRAM DOES NOT TEST DATA TRANSFERS THROUGH EITHER PORT, DOES NOT TEST THE DYNAMIC OPERATION OF THE DUAL CONTROLLER OPTION, AND DOES NOT TEST THE UNLOAD COMMAND OR THE OPERATION OF THE CONTROLLER SELECT SWITCH ON THE DRIVE. (REFER TO PARAGRAPH 2.2 & 2.3)

7.3 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 45 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 2.5 MINUTES.

7.4 REQUIRED TESTS

IF THE PROGRAM IS BEING EXECUTED IN SINGLE TEST MODE, THE OPERATOR MUST CALL AND RUN THE FOLLOWING TESTS BEFORE OTHER TESTS ARE RUN:

- A. TEST 2 AND TEST 3. THESE TESTS DETERMINE AND STORE FOR LATER USE THE TIMEOUT NON-SHOT VALUE MEASURED THROUGH EACH PORT.

7.5 DISK SURFACE USAGE

THIS DIAGNOSTIC DOES NOT USE THE DISK SURFACE. HOWEVER, THE DRIVE MUST BE CYCLED UP AND BE ON LINE FOR THE DIAGNOSTIC TO BE RUN.

7.6 LOOP ON ERROR OPTION



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

IF SW<09> IS SET, THE PROGRAM WILL LOOP ON A FAILING TEST UNTIL EITHER THE SWITCH IS RESET OR THE ERROR STOPS OCCURING. BECAUSE THE PROGRAM MUST RESET THE RM05/3/2 TO A KNOWN STATE BEFORE LOOPING ON THE ERROR, THE TEST FOR SW<09> IS PERFORMED AT THE END OF THE TEST - NOT AT THE POINT WHERE THE ERROR WAS DETECTED.

## 8. TEST DESCRIPTIONS

-----

### 8.1 METHOD USED TO VERIFY THAT THE DRIVE IS IN NEUTRAL

THE PROGRAM DETERMINES THAT THE DRIVE IS IN NEUTRAL BY CHECKING THE CONTENTS OF THE DRIVE STATUS REGISTER (RMDS) THROUGH BOTH PORTS. THE PROGRAM MASKS OUT THE PORT DEPENDENT BITS ('ATA' & 'VV') AND VERIFIES THAT CORRECT STATUS IS READ THROUGH BOTH PORTS. (THE CORRECT STATUS IS 'MOL', 'PGM', 'DPR', & 'DRY'.) IF NEITHER PORT SEES ALL ZEROS FROM RMDS, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DESCREPNACY BETWEEN PORTS INDICATES A FAILURE IN THE PATH FOR THAT BIT.

ADDITIONALLY, THE PORT REQUEST FLOPS (RQA, RQB) OF THE MAINTENANCE REGISTER ARE TESTED, AND SHOULD BE ZERO IF THE DRIVE IS IN NEUTRAL.

### 8.2 METHOD USED TO VERIFY THAT THE DRIVE HAS BEEN SEIZED

THE PROGRAM VERIFIES THAT THE DRIVE HAS BEEN SEIZED BY CHECKING THE DRIVE STATUS REGISTER (RMDS) THROUGH THE SEIZING PORT AND VERIFYING THAT CORRECT STATUS IS SEEN. WHEN RMDS IS READ THROUGH THE OPPOSITE PORT, ZEROS SHOULD BE SEEN. IF BOTH CONDITIONS EXIST, (I.E., CORRECT STATUS THROUGH THE SEIZING PORT AND ZEROS THROUGH THE OPPOSITE PORT), THE PROGRAM CONCLUDES THAT THE DRIVE HAS BEEN SEIZED BY THE SPECIFIED PORT.

### 8.3 METHOD USED TO VERIFY PORT REQUESTS

THE PORT REQUEST FLOPS IN THE MAINTENANCE REGISTER ARE TESTED TO DETERMINE IF :

- . A DRIVE IS IN NEUTRAL, I.E., RQA AND RQB ARE ZERO;
- . A DRIVE IS SEIZED, I.E., RQA OR RQB IS ONE;
- . A PORT REQUEST IS SET WHILE THE DRIVE IS SEIZED TO THE ALTERNATE PORT, I.E., RQA AND RQB ARE ONE.

## TEST 1 NEUTRAL ACCESS TEST

### VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS

- A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT THE DRIVE IS A DUAL PORT RM05/3/2, THAT THE DRIVE IS ONLINE (RMDS HAS



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  
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

'MOL', 'PGM', 'DPR', & 'DRY' BITS SET), AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS THE SAME.

B. THE TEST IS REPEATED THROUGH BOTH PORTS.

TEST 2 PORT 'A' SEIZE/TIMEOUT TEST

VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT IT CAN BE RELEASED BY THE ONE SECOND TIMER.

- A. WRITE 0'S INTO RMDA THROUGH PORT 'A'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B'; VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
- C. WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE. MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.

TEST 3 PORT 'B' SEIZE/TIMEOUT TEST

VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT IT CAN BE RELEASED BY THE ONE SECOND TIMER.

- A. WRITE 0'S INTO RMDA THROUGH PORT 'B'; VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'A'; VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
- C. WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE. MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS >500 MS.

TEST 4 PORT 'A' SEIZE/RELEASE TEST

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDA.
- B. SET VOLUME VALID AND CLEAR ANY ERROR
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.

TEST 5 PORT 'B' SEIZE/RELEASE TEST



400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
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

TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. SET VOLUME VALID AND CLEAR ANY ERROR
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE DRIVE.

TEST 6 PORT 'A' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

- A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.

TEST 7 PORT 'B' NEUTRAL/RELEASE TEST

TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL

- A. ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.

TEST 10 PORT 'A' RELEASE INTERFERENCE TEST

VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE IS SEIZED BY THE OTHER PORT.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
- C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'.
- E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 11 PORT 'B' RELEASE INTERFERENCE TEST

VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE IS SEIZED BY THE OTHER PORT.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
- C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.



457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513

- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'.
- E. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 12 PORT 'A' RELEASE W/ERRORS TEST

VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR BITS ARE SET IN THE DRIVE.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- B. WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RMER1 HAS NOT BEEN CLEARED.
- D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 13 PORT 'B' RELEASE W/ERRORS TEST

VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR BITS ARE SET IN THE DRIVE.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RMER1 HAS NOT BEEN CLEARED.
- D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
- E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 14 PORT 'A' SEIZE AND CLEAR TEST

VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING PORT TO RELEASE THE DRIVE.

- A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.



514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570

- C. ISSUE A MASSBUS CLEAR THROUGH THE RH AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 15 PORT 'B' SEIZE AND CLEAR TEST

VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING PORT TO RELEASE THE DRIVE.

- A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDs THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- C. ISSUE A MASSBUS CLEAR THROUGH THE RH AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 16 SEIZE 'A' BY RMCS1 TEST

VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE IF THE DRIVE IS IN NEUTRAL.

- A. READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'A'; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 17 SEIZE 'B' BY RMCS1 TEST

VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE IF THE DRIVE IS IN NEUTRAL.

- A. READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST

VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT REQUEST' IF THE DRIVE IS SEIZED.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.



571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627

- B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT SET.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 21 PORT 'B' INHIBIT SEIZE BY RMCS1 TEST

VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT REQUEST' IF THE DRIVE IS SEIZED.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1. VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- B. READ THE CONTROL REGISTER FROM PORT 'B'. VERIFY THAT 'DVA' IS NOT SET.
- C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 22 SEIZE BY RMAS TEST

TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER (RMAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER PORT.

- A. WRITE THE APPROPRIATE DRIVE BIT INTO RMAS; VERIFY THAT THE DRIVE IS SEIZED.
- B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.

TEST 23 INHIBIT SEIZE BY RMAS TEST

VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO THE DRIVE'S ATTENTION BIT.

- A. SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
- B. VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.

TEST 24 SET PORT 'A' REQUEST TEST

VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE DRIVE IS SEIZED BY THE OTHER PORT.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDs.
- B. WRITE 0'S INTO RMDs FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL



628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
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

SEIZED BY PORT 'B'.

- C. ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'A' AND IS NOT SET FOR PORT 'B'.
- D. ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 25 SET PORT 'B' REQUEST TEST

VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE DRIVE IS SEIZED BY THE OTHER PORT.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDs.
- B. WRITE 0'S INTO RMDs FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'.
- D. ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 26 TEST RESET ATTENTION 'A' BY DRIVE CLEAR

VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDs.
- C. ISSUE A DRIVE CLEAR COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

TEST 27 TEST RESET ATTENTION 'B' BY DRIVE CLEAR

VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS SET.



685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
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

- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- C. ISSUE A DRIVE CLEAR COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

TEST 30 RESET ATTENTION 'A' BY GO TEST

VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH ATTENTION BITS ARE SET.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- C. ISSUE A NOP COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS RESET, AND THE ATTENTION BIT FOR PORT 'B' IS STILL SET.

TEST 31 RESET ATTENTION 'B' BY GO TEST

VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE SEIZING PORT.

- A. SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH ATTENTION BITS ARE SET.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- C. ISSUE A NOP COMMAND.
- D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS RESET, AND THE ATTENTION BIT FOR PORT 'A' IS STILL SET.

TEST 32 TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT

VERIFY THAT MASSBUS CLEAR RESETS BOTH PORT'S ATTENTION BITS WHEN THE DRIVE IS IN NEUTRAL.

- A. SET THE ATTENTION BITS FOR BOTH PORTS.
- B. VERIFY THAT THE DRIVE IS IN NEUTRAL.
- C. ISSUE A MASSBUS INIT. VERIFY THAT BOTH ATTENTION BITS HAVE



742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798

RESET.

TEST 33 RESET ATTENTION 'A' & 'B' BY RMAS

VERIFY THAT BOTH ATTENTION BITS CAN BE RESET BY WRITING THE APPROPRIATE BIT IN THE ATTENTION SUMMARY REGISTER.

- A. SET THE ATTENTION BITS FOR BOTH PORTS.
- B. VERIFY THE DRIVE IS IN NEUTRAL.
- C. WRITE THE DRIVE'S ATTENTION BIT IN RMAS. VERIFY THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMAS.

TEST 34 PORT 'A' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'A'.
- B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST

VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.

- A. SET THE ATTENTION BIT FOR PORT 'B'.
- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT FOR THE DRIVE IS SET.

TEST 36 SET ATTENTION 'A' BY COMMAND TEST

TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A COMMAND.

- A. ISSUE A OFFSET COMMAND THROUGH PORT 'A'.
- B. WAIT FOR THE OFFSET COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
- C. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.



799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
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

TEST 37 SET ATTENTION 'B' BY COMMAND TEST

TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A COMMAND.

- A. ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
- B. WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME '1'). VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
- C. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

VERIFY THAT A CHANGE IN UNIT READY SETS THE ATTENTION FOR BOTH PORTS.

THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.

VERIFY THAT ATTENTION SETS WHEN THE DRIVE SWITCHES AFTER BEING RELEASED.

THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"

TEST 40 PORT 'A' SET VOLUME VALID TEST

VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.

- A. WITH PORT 'A' SELECTED, RESET AND SET 'UNIT READY' STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE IS SEIZED AND THAT 'VOLUME VALID' IS RESET AND ATTENTION IS SET.
- B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A. VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID IS SET.
- C. RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR PORT 'B'. VERIFY THAT ATTENTION IS STILL SET AND THAT VOLUME VALID IS STILL RESET.
- D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT 'B' THEN RELEASE PORT 'B'.

TEST 41 PORT 'B' SET VOLUME VALID TEST

VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.

- A. WITH PORT 'B' SELECTED, RESET AND SET 'UNIT READY' STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE



856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
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

IS SEIZED AND THAT "VOLUME VALID" IS RESET AND ATTENTION IS SET.

- B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A. VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID IS SET.
- C. RELEASE THE DRIVE FROM PORT 'B' AND SELECT THE DRIVE FOR PORT 'A'. VERIFY THAT ATTENTION IS STILL SET AND THAT VOLUME VALID IS STILL RESET.
- D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT 'A' THEN RELEASE PORT 'A'.

TEST 42 TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- B. WRITE 1'S INTO RMER1 THROUGH PORT 'A' TO FORCE AN ATTENTION.
- C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND NOT SET FOR PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

TEST 43 TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE

VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
- C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

TEST 44 PORT 'A' RETRIGGER BY DEMAND TEST

VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- B. WAIT 500 MS AND READ RMDS THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
- D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.



913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
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

TEST 45 PORT 'B' RETRIGGER BY DEMAND TEST

VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. WAIT 500 MS AND WRITE 0'B INTO RMDS THROUGH PORT 'A'.
- C. VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
- D. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 46 PORT 'A' TIMEOUT/RELEASE TEST

VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'A'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.
- E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 47 PORT 'B' TIMEOUT/RELEASE TEST

VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'B'.
- C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
- D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.

TEST 50 PORT 'A' SEIZE ACCESS TEST



970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDs.
- B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'.
- C. READ RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RMER1, RMER2 THROUGH PORT 'A'.
- E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
- F. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'B' AND THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
- G. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

TEST 51 PORT 'B' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDs.
- B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.
- C. READ RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'A' SEES 0'S FROM EACH OF THESE REGISTERS.
- D. CLEAR RMER1, RMER2 THROUGH PORT 'B'.
- E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
- F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
- G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.



1  
606  
607

```
;PROGRAM REVISION #001
.TITLE CZRMRAO RM05/3/2 DU POR TST 1
;*COPYRIGHT (C) 1980
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY MIKE LEAVITT
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C4), 1980.
```

608

```
.SBTTL OPERATIONAL SWITCH SETTINGS
;*
;* SWITCH USE
;* -----
;* 15 HALT ON ERROR
;* 14 LOOP ON TEST
;* 13 INHIBIT ERROR TYPEOUTS
;* 11 INHIBIT ITERATIONS
;* 10 BELL ON ERROR
;* 9 LOOP ON ERROR
```

609  
610

```
.SBTTL BASIC DEFINITIONS
;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100 STACK = 1100
104000 ERROR = EMT ;;BASIC DEFINITION OF ERROR CALL
000004 SCOPE = IOT ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS
000011 HT = 11 ;;CODE FOR HORIZONTAL TAB
000012 LF = 12 ;;CODE FOR LINE FEED
000015 CR = 15 ;;CODE FOR CARRIAGE RETURN
000200 CRLF = 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776 PS = 177776 ;;PROCESSOR STATUS WORD
177776 PSW=PS
177774 STKLMT = 177774 ;;STACK LIMIT REGISTER
177772 PIRQ = 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
177570 DSWR = 177570 ;;HARDWARE SWITCH REGISTER
177570 DDISP = 177570 ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS
000000 R0 = %0 ;;GENERAL REGISTER
000001 R1 = %1 ;;GENERAL REGISTER
000002 R2 = %2 ;;GENERAL REGISTER
000003 R3 = %3 ;;GENERAL REGISTER
000004 R4 = %4 ;;GENERAL REGISTER
000005 R5 = %5 ;;GENERAL REGISTER
000006 R6 = %6 ;;GENERAL REGISTER
000007 R7 = %7 ;;GENERAL REGISTER
000006 SP = %6 ;;STACK POINTER
000007 PC = %7 ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
000000 PRO = 0 ;;PRIORITY LEVEL 0
000040 PR1 = 40 ;;PRIORITY LEVEL 1
```



BASIC DEFINITIONS

000100	PR2	=	100	::	PRIORITY LEVEL 2
000140	PR3	=	140	::	PRIORITY LEVEL 3
000200	PR4	=	200	::	PRIORITY LEVEL 4
000240	PR5	=	240	::	PRIORITY LEVEL 5
000300	PR6	=	300	::	PRIORITY LEVEL 6
000340	PR7	=	340	::	PRIORITY LEVEL 7

;'SWITCH REGISTER' SWITCH DEFINITIONS

100000	SW15	=	100000
040000	SW14	=	40000
020000	SW13	=	20000
010000	SW12	=	10000
004000	SW11	=	4000
002000	SW10	=	2000
001000	SW09	=	1000
000400	SW08	=	400
000200	SW07	=	200
000100	SW06	=	100
000040	SW05	=	40
000020	SW04	=	20
000010	SW03	=	10
000004	SW02	=	4
000002	SW01	=	2
000001	SW00	=	1
001000	SW9=SW09		
000400	SW8=SW08		
000200	SW7=SW07		
000100	SW6=SW06		
000040	SW5=SW05		
000020	SW4=SW04		
000010	SW3=SW03		
000004	SW2=SW02		
000002	SW1=SW01		
000001	SW0=SW00		

;'DATA BIT DEFINITIONS (BIT00 TO BIT15)

100000	BIT15	=	100000
040000	BIT14	=	40000
020000	BIT13	=	20000
010000	BIT12	=	10000
004000	BIT11	=	4000
002000	BIT10	=	2000
001000	BIT09	=	1000
000400	BIT08	=	400
000200	BIT07	=	200
000100	BIT06	=	100
000040	BIT05	=	40
000020	BIT04	=	20
000010	BIT03	=	10
000004	BIT02	=	4
000002	BIT01	=	2
000001	BIT00	=	1
001000	BIT9=BIT09		
000400	BIT8=BIT08		
000200	BIT7=BIT07		
000100	BIT6=BIT06		
000040	BIT5=BIT05		



```

000020 BIT4=BIT04
000010 BIT3=BIT03
000004 BIT2=BIT02
000002 BIT1=BIT01
000001 BIT0=BIT00

;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004 ERRVEC = 4 ;:TIME OUT AND OTHER ERRORS
000010 RESVEC = 10 ;:RESERVED AND ILLEGAL INSTRUCTIONS
000014 TBITVEC = 14 ;: "T" BIT
000014 TRTVEC = 14 ;:TRACE TRAP
000014 BPTVEC = 14 ;:BREAKPOINT TRAP (BPT)
000020 IOTVEC = 20 ;:INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024 PWRVEC = 24 ;:POWER FAIL
000030 EMTVEC = 30 ;:EMULATOR TRAP (EMT) **ERROR**
000034 TRAPVEC = 34 ;:"TRAP" TRAP
000060 TKVEC = 60 ;:TTY KEYBOARD VECTOR
000064 TPVEC = 64 ;:TTY PRINTER VECTOR
000240 PIRQVEC = 240 ;:PROGRAM INTERRUPT REQUEST VECTOR
    
```

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  
648

.SBTTL RH11/RH70 REGISTERS

;CONTROL AND STATUS REGISTER 1 (RMCS1)

```

000100 IE = 100 ;:INTERRUPT ENABLE (BIT #6)
000200 RDY = 200 ;:READY (BIT #7)
000400 A16 = 400 ;:HIGH ORDER BUS ADDRESS BIT (BIT #8)
001000 A17 = 1000 ;:HIGH ORDER BUS ADDRESS BIT (BIT #9)
002000 PSEL = 2000 ;:PORT SELECT (BIT #10)
020000 MCPE = 20000 ;:MASSBUS PARITY ERROR (BIT #13)
040000 TRE = 40000 ;:TRANSFER ERROR (BIT #14)
100000 SC = 100000 ;:SPECIAL CONDITION (BIT #15)
    
```

;CONTROL AND STATUS REGISTER 2 (RMCS2)

```

000001 U0 = 1 ;:UNIT SELECT (BIT #0)
000002 U1 = 2 ;:UNIT SELECT (BIT #1)
000004 U3 = 4 ;:UNIT SELECT (BIT #2)
000010 BAI = 10 ;:BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020 PAT = 20 ;:MASSBUS PARITY TEST (BIT #4)
000040 CLR = 40 ;:CLEAR (BIT #5)
000100 IR = 100 ;:INPUT READY (BIT #6)
000200 OR = 200 ;:OUTPUT READY (BIT #7)
000400 MDPE = 400 ;:MASS BUS PARITY ERROR (BIT #8)
001000 MXF = 1000 ;:MISSED TRANSFER ERROR (BIT #9)
002000 PGE = 2000 ;:PROGRAM ERROR (BIT #10)
004000 NEM = 4000 ;:NON EXISTENT MEMORY (BIT #11)
010000 NED = 10000 ;:NON EXISTENT DRIVE (BIT #12)
020000 UPE = 20000 ;:UNIBUS PARITY ERROR (BIT #13)
040000 WCE = 40000 ;:WRITE CHECK ERROR (BIT #14)
100000 DLT = 100000 ;:DATA LATE (BIT #15)
    
```

;DATA BUFFER REGISTER (RMDB)  
 ;(EACH BIT IS CALLED BY BIT NUMBER)

.SBTTL RM REGISTERS



```

649
650           ;CONTROL AND STATUS REGISTER #1. (#00)
651           000001      GO      = 1           ;GO BIT (BIT #0)
652           000002      F0      = 2           ;FUNCTION CODE BIT #1
653           000004      F1      = 4           ;FUNCTION CODE BIT #2
654           000010      F2      = 10          ;FUNCTION CODE BIT #3
655           000020      F3      = 20          ;FUNCTION CODE BIT #4
656           000040      F4      = 40          ;FUNCTION CODE BIT #5
657           004000      DVA     = 4000        ;DEVICE AVAILABLE (BIT #11)
658
659           ;CONTROL STATUS REGISTER #2 (RMCS2)
660
661           000040      CLR     = BITS        ;CONTROLLER CLEAR
662
663           ;DRIVE STATUS REGISTER (RMDS) (#01)
664
665           000001      OM      = BIT00        ;OFFSET MODE
666           000100      VV      = 100         ;VOLUME VALID (BIT #6)
667           000200      DRY     = 200         ;DRIVE READY (BIT #7)
668           000400      DPR     = 400         ;DRIVE PRESENT (BIT #8)
669           001000      PGM     = 1000        ;PROGRAMABLE (BIT #9)
670           002000      LBT     = 2000        ;LAST SECTOR TRANSFERRED (BIT #10)
671           004000      WRL     = 4000        ;WRITE LOCK (BIT #11)
672           010000      MOL     = 10000       ;MEDIUM ON-LINE (BIT #12)
673           020000      PIP     = 20000      ;POSITIONING OPERATION IN PROGRESS (BIT #13)
674           040000      ERR     = 40000      ;COMPOSITE ERROR (BIT #14)
675           100000      ATA     = 100000     ;ATTENTION ACTIVE (BIT #15)
676
677           ;ERROR REGISTER #01 (RMER1) (#02)
678
679           000001      ILF     = 1           ;ILLEGAL FUNCTION (BIT #0)
680           000002      ILR     = 2           ;ILLEGAL REGISTER (BIT #1)
681           000004      RMR     = 4           ;REGISTER MODIFICATION REFUSED (BIT #2)
682           000010      PAR     = 10          ;PARITY ERROR (BIT #3)
683           000020      FER     = 20          ;FORMAT ERROR (BIT #4)
684           000040      WCF     = 40          ;WRITE CLOCK FAIL (BIT #5)
685           000100      ECH     = 100         ;ECC HARD ERROR (BIT #6)
686           000200      HCE     = 200         ;HEADER COMPARE ERROR (BIT #7)
687           000400      HCRC    = 400         ;HEADER CRC ERROR (BIT #8)
688           001000      AOE     = 1000        ;ADDRESS OVERFLOW ERROR (BIT #9)
689           002000      IAE     = 2000        ;INVALID ADDRESS ERROR (BIT #10)
690           004000      WLE     = 4000        ;WRITE LOCK ERROR (BIT #11)
691           010000      DTE     = 10000       ;DRIVE TIMING ERROR (BIT #12)
692           020000      OPI     = 20000      ;OPERATION INCOMPLETE (BIT #13)
693           040000      UNS     = 40000      ;DRIVE UNSAFE (BIT #14)
694           100000      DCK     = 100000     ;DATA CHECK ERROR (BIT 15)
695
696           ;MAINTAINABILITY REGISTER (RMMR1)(#03)
697
698           000001      DMD     = 1           ;DIAGINOSTIC MODE (BIT #0)
699           001000      MUR     = BIT09       ;MAINTENANCE UNIT READY
700           040000      RQB     = BIT14       ;PORT B REQUEST FLOP
701           100000      RQA     = BIT15       ;PORT A REQUEST FLOP
702
703           ;ATTENTION SUMMARY PSEUDO-REGISTER (RMAS) (#04)
704
705           000001      ATO     = 1           ;DEVICE 0 (BIT #0)

```



```

RM REGISTERS

706      000002      AT1      = 2      ;DEVICE 1 (BIT #1)
707      000004      AT2      = 4      ;DEVICE 2 (BIT #2)
708      000010      AT3      = 10     ;DEVICE 3 (BIT #3)
709      000020      AT4      = 20     ;DEVICE 4 (BIT #4)
710      000040      AT5      = 40     ;DEVICE 5 (BIT #5)
711      000100      AT6      = 100    ;DEVICE 6 (BIT #6)
712      000200      AT7      = 200    ;DEVICE 7 (BIT #7)
713
714      ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RMDA) (#05)
715      ;(EACH BIT IS CALLED BY BIT NUMBER)
716
717      ;DRIVE TYPE REGISTER (RMDT) (#06)
718
719      000001      DT00     = 1      ;DRIVE TYPE NUMBER BIT 1
720      000002      DT01     = 2      ;DRIVE TYPE NUMBER BIT 2
721      000004      DT02     = 4      ;DRIVE TYPE NUMBER BIT 3
722      000010      DT03     = 10     ;DRIVE TYPE NUMBER BIT 4
723      000020      DT04     = 20     ;DRIVE TYPE NUMBER BIT 5
724      000040      DT05     = 40     ;DRIVE TYPE NUMBER BIT 6
725      000100      DT06     = 100    ;DRIVE TYPE NUMBER BIT 7
726      000200      DT07     = 200    ;DRIVE TYPE NUMBER BIT 8
727      000400      DT08     = 400    ;DRIVE TYPE NUMBER BIT 9
728      004000      DRQ      = 4000   ;DRIVE REQUEST REQUIRED (BIT #11)
729      020000      MOH      = 20000  ;MOVING HEAD (BIT #13)
730      040000      TAP      = 40000  ;TAPE DRIVE (BIT #14)
731      100000      NBA      = 100000  ;NOT BLOCK ADDRESSED (BIT #15)
732
733      ;LOOK-AHEAD REGISTER (RMLA) (#07)
734
735      000100      SC0      = 100    ;SECTOR COUNT FIELD 0 (BIT #6)
736      000200      SC1      = 200    ;SECTOR COUNT FIELD 1 (BIT #7)
737      000400      SC2      = 400    ;SECTOR COUNT FIELD 2 (BIT #8)
738      001000      SC3      = 1000   ;SECTOR COUNT FIELD 3 (BIT #9)
739      002000      SC4      = 2000   ;SECTOR COUNT FIELD 4 (BIT #10)
740
741      ;RM ERROR REGISTER #2 (RMER2) (#10)
742
743      000010      DPE      = 10     ;DATA PARITY ERROR (BIT #3)
744      000200      DVC      = 200    ;DEVICE CHECK (BIT #7)
745      002000      LBC      = 2000   ;LOSS OF BIT CLOCK (BIT #10)
746      004000      LSC      = 4000   ;LOSS OF SYSTEM CLOCK (BIT #11)
747      010000      IVC      = 10000  ;INVALID COMMAND (BIT #12)
748      020000      OPE      = 20000  ;OPERATOR ERROR (BIT #13)
749      100000      SKI      = 100000 ;SEEK INCOMPLETE (BIT #14)
750
751      ;OFFSET REGISTER (RMOF) (#11)
752
753      000200      OFD      = 200    ;OFFSET FORWARD (BIT #5)
754      002000      HCI      = 2000   ;HEADER COMPARE INHIBIT (BIT #10)
755      004000      ECI      = 4000   ;ERROR CORRECTION CODE INHIBIT (BIT #11)
756      010000      FMT16   = 10000  ;FORMAT BIT (BIT #12)
757
758      ;DESIRED CYLINDER ADDRESS (RMDC) (#12)
759      ;(EACH BIT IS CALLED BY BIT NUMBER)
760
761      ;SERIAL NUMBER REGISTER (RMSN) (#14)
762      ;(EACH IS CALLED BY BIT NUMBER)

```



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  
793  
794

;ECC POSITION REGISTER (RMEC1) (#16)  
;(EACH BIT IS CALLED BY BIT NUMBER)

;ECC PATTERN REGISTER (RMEC2) (#17)  
;(EACH BIT IS CALLED BY BIT NUMBER)

.SBTTL DEFINITIONS OF THE RH/RM ADDRESS INDEXES

000000	RMCS1	= 0	;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
000002	RMWC	= 2	;WORD COUNT REGISTER (NOT A DRIVE REG)
000004	RMBA	= 4	;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
000006	RMDA	= 6	;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
000010	RMCS2	= 10	;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
000012	RMDS	= 12	;DRIVE STATUS REGISTER (DRIVE REG 01)
000014	RMER1	= 14	;ERROR REGISTER #1 (DRIVE REG. 02)
000016	RMAS	= 16	;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
000020	RMLA	= 20	;LOOK AHEAD REGISTER (DRIVE REG. 07)
000022	RMDB	= 22	;DATA BUFFER REGISTER (NOT A DRIVE REG.)
000024	RMMR1	= 24	;MAINTAINABILITY REGISTER (DRIVE REG. 03)
000026	RMDT	= 26	;DRIVE TYPE REGISTER (DRIVE REG. 06)
000030	RMSN	= 30	;SERIAL NUMBER REGISTER (DRIVE REG. 10)
000032	RMOF	= 32	;OFFSET REGISTER (DRIVE REG. 11)
000034	RMDC	= 34	;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
000040	RMMR2	= 40	;MAINTENANCE REGISTER #2 (DRIVE REG. 14)
000042	RMER2	= 42	;ERROR REGISTER #2 (DRIVE REG. 15)
000044	RMEC1	= 44	;ECC POSITION REGISTER (DRIVE REG. 16)
000046	RMEC2	= 46	;ECC PATTERN REGISTER (DRIVE REG. 17)

.SBTTL TRAP CATCHER

000000  
.=0  
;\*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"  
;\*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS  
;\*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

000174	000174	000000	DISPREG: .WORD 0	::SOFTWARE DISPLAY REGISTER
000176	000000	000000	SWREG: .WORD 0	::SOFTWARE SWITCH REGISTER

.SBTTL STARTING ADDRESS(ES)

795	000200	000137	002240	JMP @#START	::JUMP TO STARTING ADDRESS OF PROGRAM
796	000204	000137	002246	JMP @#START1	;START AND CHANGE THE RH/RM ADDRESS

.SBTTL ACT11 HOOKS

```

;*****
;HOOKS REQUIRED BY ACT11
    000210    $SVPC=.           ;SAVE PC
    000046    .=46
    066104    $ENDAD           ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
    000052    .=52
    020000    .WORD 20000      ;;2)SET LOC.52 TO 20000
    000210    .=$SVPC         ;;RESTORE PC
    
```

799



840







0

.SBTTL USER DEFINED TAGS

001212	172540	\$LKCSR: .WORD	172540	; ADDR OF KW11-P STATUS REGISTER
001214	172542	\$LKCSB: .WORD	172542	; ADDR OF KW11-P COUNTER BUFFER
001216	000104	\$LPVEC: .WORD	104	; ADDR OF KW11-P VECTOR
001220	177546	\$LKS: .WORD	177546	; ADDR OF KW11-L STATUS REGISTER
001222	000100	\$LLVEC: .WORD	100	; ADDR OF KW11-L VECTOR
001224	000000	PORTA: .WORD	0	; ADDRESS OF PORT A
001226	000000	PORTB: .WORD	0	; ADDRESS OF PORT B
001230	000000	PORTC: .WORD	0	; ADDRESS OF DIFFERENT DRIVE
001232	000000	RQSTA: .WORD	0	; REQUEST BIT FOR PORT A
001234	000000	RQSTB: .WORD	0	; REQUEST BIT FOR PORT B
001236	000000	ASR1: .WORD	0	; ATA-A OR ATA-B = 1
001240	000000	PTNBR: .WORD	0	; CONTAINS THE PORT ADDRESS FOR ERROR TYPEOUTS
001242	000000	SEIZPT: .WORD	0	; CONTAINS THE ADDRESS OF THE SEIZING PORT
001244	000000	OPPR: .WORD	0	; CONTAINS THE ADDRESS OF THE 'OPPOSITE' PORT
001246	000000	TSTNUM: .WORD	0	; NUMBER OF THE CURRENT TEST
001250	000000	CKERR: .WORD	0	; IF -1, A REGISTER MISCOMPARISON OCCURRED
001252	000000	NOSEIZ: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT SEIZE THE DRIVE
001254	000000	RELERR: .WORD	0	; IF -1, THE PORT IN 'SEIZPT' DID NOT RELEASE THE DRIVE
001256	000000	TIME: .WORD	0	; ELAPSED TIME COUNTER
001260	000000	WATCH: .WORD	0	; WATCH DOG TIMER LOCATION
001262	000000	TIMEA: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT A
001264	000000	TIMEAP: .WORD	0	; PORT A TIMEOUT VALUE + 25%
001266	000000	TIMEAM: .WORD	0	; PORT A TIMEOUT VALUE - 25%
001270	000000	TIMEB: .WORD	0	; THE TIMEOUT ONE-SHOT VALUE MEASURED THROUGH PORT B
001272	000000	TIMEBP: .WORD	0	; PORT B TIMEOUT VALUE + 25%
001274	000000	TIMEBM: .WORD	0	; PORT B TIME VALUE - 25%
001276	000000	TIMES: .WORD	0	; STORAGE FOR TIMEOUT ONE-SHOT RETRIGGER TEST
001300	000000	KYBCTL: .WORD	0	; SINGLE TEST INDICATOR
001302	000000	CHGADR: .WORD	0	; CHANGE THE RH/RM ADDRESS INDICATOR

.SBTTL RH/RM UNIBUS AND VECTOR ADDRESSES

001304	176700	\$RMADR: .WORD	176700	; RH/RM UNIBUS ADDRESS
001306	000254	\$RMVEC: .WORD	254	; INTERRUPT VECTOR ADDRESS



0

.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

1	001310		\$ERRTB:		
2			;ERROR 1		
3					
4	001310	072076		EM1	;WRONG DRIVE TYPE
5	001312	076520		DH1	
6	001314	100410		DT1	
7	001316	100676		DF1	
8					
9			;ERROR 2		
10					
11	001320	072117		EM2	;DRIVE NOT ON LINE
12	001322	076520		DH1	
13	001324	100410		DT1	
14	001326	100676		DF1	
15					
16			;ERROR 3		
17					
18	001330	072141		EM3	;SERIAL NUMBERS NOT THE SAME
19	001332	076571		DH3	
20	001334	100424		DT3	
21	001336	100676		DF1	
22					
23			;ERROR 4		
24					
25	001340	072223		EM4	;DRIVE NOT SEIZED BY PORT 'N'
26	001342	076640		DH4	
27	001344	100472		DT7	
28	001346	100711		DF7	
29					
30			;ERROR 5		
31					
32	001350	072254		EM5	;WRONG STATUS SEEN BY THE SEIZING PORT
33	001352	076763		DH5	
34	001354	100440		DT5	
35	001356	100703		DF5	
36					
37			;ERROR 6		
38					
39	001360	072322		EM6	;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED
40	001362	077233		DH13	
41	001364	100512		DT13	
42	001366	100703		DF5	



43				
44			:ERROR 7	
45				
46	001370	072422	EM7	:REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
47	001372	077037	DH7	
48	001374	100472	DT7	
49	001376	100711	DF7	
50				
51			:ERROR 10	
52				
53	001400	072503	EM10	:REGISTER CONTENTS INCORRECT
54	001402	076763	DH5	
55	001404	100440	DT5	
56	001406	100703	DF5	
57				
58			:ERROR 11	
59				
60	001410	072533	EM11	:CONTROL BUS PARITY ERROR WHILE READING REGISTER
61	001412	077162	DH11	
62	001414	100410	DT1	
63	001416	100676	DF1	
64				
65			:ERROR 12	
66				
67	001420	072617	EM12	:DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
68	001422	077727	DH36	
69	001424	100600	DT37	
70	001426	100724	DF36	
71				
72			:ERROR 13	
73				
74	001430	072667	EM13	: 'VOLUME VALID' BIT NOT SET BY READIN PRESET
75	001432	077233	DH13	
76	001434	100512	DT13	
77	001436	100703	DF5	
78				
79			:ERROR 14	
80				
81	001440	072754	EM14	: 'VOLUME VALID' SET ON THE OPPOSITE PORT
82	001442	077233	DH13	
83	001444	100512	DT13	
84	001446	100703	DF5	
85				
86			:ERROR 15	
87				
88	001450	073017	EM15	:THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
89	001452	077037	DH7	
90	001454	100472	DT7	
91	001456	100711	DF7	
92				
93			:ERROR 16	
94				
95	001460	073076	EM16	:ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
96	001462	077037	DH7	
97	001464	100472	DT7	
98	001466	100711	DF7	
99				

Line	Code	Pointer	Label	Description
100			;ERROR 17	
101				
102	001470	073151	EM17	;ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
103	001472	077037	DH7	
104	001474	100472	DT7	
105	001476	100711	DF7	
106				
107			;ERROR 20	
108				
109	001500	073230	EM20	;DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
110	001502	077727	DH36	
111	001504	100600	DT37	
112	001506	100724	DF36	
113				
114			;ERROR 21	
115				
116	001510	073310	EM21	;DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
117	001512	077727	DH36	
118	001514	100600	DT37	
119	001516	100724	DF36	
120				
121			;ERROR 22	
122				
123	001520	073363	EM22	;DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
124	001522	077353	DH22	
125	001524	100530	DT22	
126	001526	100720	DF31	
127				
128			;ERROR 23	
129				
130	001530	073450	EM23	;TIMEOUT CLEARED THE DRIVE'S ERROR BIT
131	001532	077451	DH23	
132	001534	100542	DT23	
133	001536	100676	DF1	
134				
135			;ERROR 24	
136				
137	001540	073516	EM24	;RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
138	001542	077451	DH23	
139	001544	100542	DT23	
140	001546	100676	DF1	
141				
142				
143			;ERROR 25	
144				
145	001550	073575	EM25	;TIMEOUT ONE-SHOT DID NOT RETRIGGER
146	001552	077727	DH36	
147	001554	100570	DT36	
148	001556	100724	DF36	
149				
150				
151			;ERROR 26	
152				
153	001560	073640	EM26	;DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
154	001562	077353	DH22	
155	001564	100530	DT22	
156	001566	100720	DF31	



157				
158				
159				
160	001570	073725		
161	001572	077037		
162	001574	100472		
163	001576	100711		
164				
165				
166				
167	001600	074003		
168	001602	077727		
169	001604	100570		
170	001606	100724		
171				
172				
173				
174	001610	074100		
175	001612	077630		
176	001614	100556		
177	001616	100720		
178				
179				
180				
181	001620	074155		
182	001622	076763		
183	001624	100440		
184	001626	100703		
185				
186				
187				
188	001630	074226		
189	001632	077727		
190	001634	100570		
191	001636	100724		
192				
193				
194				
195	001640	074330		
196	001642	077727		
197	001644	100570		
198	001646	100724		
199				
200				
201				
202	001650	074433		
203	001652	077727		
204	001654	100600		
205	001656	100724		
206				
207				
208				
209	001660	074512		
210	001662	077727		
211	001664	100570		
212	001666	100724		
213				

;ERROR 27

EM27  
 DH7  
 DT7  
 DF7

;REGISTER WRONG AFTER RELEASE WITH REQUEST SET

;ERROR 30

EM30  
 DH36  
 DT36  
 DF36

;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL

;ERROR 31

EM31  
 DH31  
 DT31  
 DF31

;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SET

;ERROR 32

EM32  
 DH5  
 DT5  
 DF5

;ATTN BIT WRONG AFTER RECALIBRATE COMMAND

;ERROR 33

EM33  
 DH36  
 DT36  
 DF36

;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED

;ERROR 34

EM34  
 DH36  
 DT36  
 DF36

;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED

;ERROR 35

EM35  
 DH36  
 DT37  
 DF36

;DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT ONE SHOT

;ERROR 36

EM36  
 DH36  
 DT36  
 DF36

;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS

Address	Code	Hex	Label	Description
214			:ERROR 37	
215				
216	001670	074564	EM37	:DRIVE IS NON-EXISTENT
217	001672	077727	DH36	
218	001674	100600	DT37	
219	001676	100724	DF36	
220				
221			:ERROR 40	
222				
223	001700	074632	EM40	:ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
224	001702	076520	DH1	
225	001704	100542	DT23	
226	001706	100676	DF1	
227				
228			:ERROR 41	
229				
230	001710	074707	EM41	:TIMEOUT CLEARED ATTENTION BIT
231	001712	077451	DH23	
232	001714	100542	DT23	
233	001716	100676	DF1	
234				
235			:ERROR 42	
236				
237	001720	074751	FM42	:DRIVE NOT IN NEUTRAL OR SEIZED
238	001722	077756	DH42	
239	001724	100610	DT42	
240	001726	100727	DF42	
241				
242			:ERROR 43	
243				
244	001730	075037	EM43	:DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
245	001732	077756	DH42	
246	001734	100610	DT42	
247	001736	100727	DF42	
248				
249			:ERROR 44	
250				
251	001740	075114	EM44	:WRITE ATTENTION BIT DID NOT SET PORT REQUEST
252	001742	077775	DH44	
253	001744	100556	DT31	
254	001746	100720	DF31	
255				
256			:ERROR 45	
257				
258	001750	075171	EM45	:CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'
259	001752	076520	DH1	
260	001754	100410	DT1	
261	001756	100676	DF1	
262				
263			:ERROR 46	
264				
265	001760	075250	EM46	:CAN'T ACCESS DRIVE THROUGH EITHER PORT
266	001762	100073	DH46	
267	001764	100616	DT46	
268	001766	100720	DF31	
269				
270			:ERROR 47	



271				
272	001770	075317	EM47	;ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
273	001772	077451	DH23	
274	001774	100542	DT23	
275	001776	100676	DF1	
276				
277				;ERROR 50
278				
279	002000	075405	EM50	;ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMAND
280	002002	077233	DH13	
281	002004	100512	DT13	
282	002006	100703	DF5	
283				
284				;ERROR 51
285				
286	002010	075467	EM51	;ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
287	002012	076763	DH5	
288	002014	100440	DT5	
289	002016	100703	DF5	
290				
291				;ERROR 52
292				
293	002020	075556	EM52	;ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
294	002022	077233	DH13	
295	002024	100512	DT13	
296	002026	100703	DF5	
297				
298				;ERROR 53
299				
300	002030	075651	EM53	;CAN'T READ ATTN BIT FROM OPPOSITE PORT
301	002032	077451	DH23	
302	002034	100410	DT1	
303	002036	100676	DF1	
304				
305				;ERROR 54
306				
307	002040	075732	EM54	;RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
308	002042	077353	DH22	
309	002044	100630	DT54	
310	002046	100720	DF31	
311				
312				;ERROR 55
313				
314	002050	076025	EM55	;TIMEOUT ONE-SHOT IS LESS THAN 500 MS
315	002052	100170	DH55	
316	002054	100642	DT55	
317	002056	100731	DF55	
318				
319				;ERROR 56
320				
321	002060	076072	EM56	;RH/RM DIDN'T RESPOND TO ADDRESSING
322	002062	100246	DH56	
323	002064	100654	DT56	
324	002066	100735	DF56	
325				
326				;ERROR 57
327				

ERROR POINTER TABLE

328				
329	002070	076135	EM57	;PORT REQUEST FLOPS WRONG
330	002072	100255	DH57	
331	002074	100660	DT57	
332	002076	100703	DF5	
333				
334				
335				;ERROR 60
336	002100	076176	EM60	;ATTENTION BITS NOT RESET BY RMAS
337	002102	076763	DH5	
338	002104	100440	DT5	
339	002106	100703	DF5	
340				
341				;ERROR 61
342				
343	002110	076242	EM61	;ATTENTION NOT RESET BY GO
344	002112	077451	DH23	
345	002114	100542	DT23	
346	002116	100676	DF1	
347				
348				;ERROR 62
349				
350	002120	076274	EM62	;ATTENTION RESET BY GO WHEN NOT SEIZED
351	002122	077233	DH13	
352	002124	100512	DT13	
353	002126	100703	DF5	
354				
355				;ERROR 63
356				
357	002130	076342	EM63	;DRIVE SEIZED BY UNIT READY CHANGE
358	002132	077727	DH36	
359	002134	100570	DT36	
360	002136	100724	DF36	
361				
362				;ERROR 64
363				
364	002140	076404	EM64	;ATTENTION NOT SET BY UNIT READY CHANGE
365	002142	077037	DH7	
366	002144	100472	DT7	
367	002146	100711	DF7	
368				
369				;ERROR 65
370				
371	002150	076453	EM65	;VV NOT RESET BY UNIT READY
372	002152	076763	DH5	
373	002154	100440	DT5	
374	002156	100703	DF5	
375				
381				



```

1          ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
2
3          BADTMO: MOV    (SP),R0          ;SAVE PC WHERE THE TIME OUT OCCURED
4          TST     -(R0)                ;ADJUST PC -2
5          CMP     (SP)+,(SP)+          ;RESTORE STACK POINTER
6          TYPE    ,65$                 ;:TYPE ASCIZ STRING
          BR      64$                   ;:GET OVER THE ASCIZ
          ;:65$: .ASCIZ <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
          64$:
7          MOV     R0,-(SP)              ;SETUP FOR TYPING OUT PC
8          TYPOC
9          NOP                            ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
          ;TO STOP ON UNEXPECTED TIMEOUT.
10
11
12         .SBTTL  START OF PROGRAM
13
14         START: CLR     CHGADR          ;CLEAR THE 'CHANGE RH/RM ADDRESS' INDICATOR
15         BR      START2                ;GO TO THE START
16
17         START1: MOV    #-1,CHGADR     ;SET THE 'CHANGE RH/RM ADDRESS' INDICATOR
18
19         START2: NOP
20         INC     #0                    ;TTY LOOP, WAIT FOR INCREMENT
21         BNE    .-4                    ;OF WORD
22         RESET
23         ;CLEAR THE WORLD
24
          .SBTTL  INITIALIZE THE COMMON TAGS
          ;:CLEAR THE COMMON TAGS (%CMTAG) AREA
          MOV     #CMTAG,R6              ;:FIRST LOCATION TO BE CLEARED
          CLR     (R6)+                  ;:CLEAR MEMORY LOCATION
          CMP     #SWR,R6                ;:DONE?
          BNE    .-6                     ;:LOOP BACK IF NO
          MOV     #STACK,SP              ;:SETUP THE STACK POINTER
          ;:INITIALIZE A FEW VECTORS
          MOV     #SCOPE,@#IOTVEC        ;:IOT VECTOR FOR SCOPE ROUTINE
          MOV     #340,@#IOTVEC+2        ;:LEVEL 7
          MOV     #ERROR,@#EMTVEC        ;:EMT VECTOR FOR ERROR ROUTINE
          MOV     #340,@#EMTVEC+2        ;:LEVEL 7
          MOV     #TRAP,@#TRAPVEC        ;:TRAP VECTOR FOR TRAP CALLS
          MOV     #340,@#TRAPVEC+2        ;:LEVEL 7
          MOV     $ENDCT,$EOPCT          ;:SETUP END-OF-PROGRAM COUNTER
          CLR     $TIMES                  ;:INITIALIZE NUMBER OF ITERATIONS
          CLR     $ESCAPE                  ;:CLEAR THE ESCAPE ON ERROR ADDRESS
          MOVB    #1,$ERMAX               ;:ALLOW ONE ERROR PER TEST
          MOV     #,$LPADR                 ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
          MOV     #,$LPERR                 ;:SETUP THE ERROR LOOP ADDRESS
          ;:SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
          ;:EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
          MOV     @#ERRVEC,-(SP)          ;:SAVE ERROR VECTOR
          MOV     #64$,@#ERRVEC           ;:SET UP ERROR VECTOR
          MOV     #DSWR,$SWR              ;:SETUP FOR A HARDWARE SWICH REGISTER
          MOV     #DDISP,$DISPLAY         ;:AND A HARDWARE DISPLAY REGISTER
          CMP     #-1,@SWR                ;:TRY TO REFERENCE HARDWARE SWR
          BNE    66$                      ;:BRANCH IF NO TIMEOUT TRAP OCCURRED
          ;:AND THE HARDWARE SWR IS NOT = -1
          BR      65$                     ;:BRANCH IF NO TIMEOUT
          64$: MOV     #65$,(SP)          ;:SET UP FOR TRAP RETURN
    
```

```

002456 000002
002460 012737 000176 001140 65$: RTI
002466 012737 000174 001142 MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
002474 012637 000004 66$: MOV #DISPREG,DISPLAY
MOV (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR

25 ;SETUP "TIMEOUT" TRAP VECTOR FOR UNEXPECTED BUS TIMEOUTS
26 002500 012737 002160 000004 MOV #BADTMO,ERRVEC ;;SETUP FOR UNEXPECTED TIMEOUT
27 002506 012737 000300 000006 MOV #PR6,ERRVEC+2 ;;LEVEL 6
28 002514 012746 000140 MOV #PR3,-(SP) ;;PUT NEW PS ON STACK
002520 012746 002526 MOV #67$,-(SP) ;;PUT NEW PC ON STACK
002524 000002 RTI ;;POP NEW PC AND PS
002526 67$:

29
30 .SBTTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
002526 005227 177777 INC #-1 ;;FIRST TIME?
002532 001054 BNE 68$ ;;BRANCH IF NO
002534 022737 066104 000042 CMP #SENDAD,@#42 ;;ACT-11?
002542 001450 BEQ 68$ ;;BRANCH IF YES
002544 104401 002602 TYPE ,69$ ;;TYPE ASCIZ STRING

.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
002550 005737 000042 TST @#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
002554 001006 BNE 70$ ;;BRANCH IF YES
002556 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
002564 001005 BNE 71$ ;;BRANCH IF NO
002566 104406 GTSWR ;;GET SOFT-SWR SETTINGS
002570 000403 BR 71$
002572 112737 000001 001134 70$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
002600 71$:
002600 000431 BR 68$ ;;GET OVER THE ASCIZ
002664 68$: .ASCIZ <CRLF>@CZRMRAO - RM05/3/2 DUAL PORT LOGIC TEST, PART 1@<CRLF>

31
32 002664 004737 070074 JSR PC,$TKINT ;;SETUP THE TTY KEYBOARD
33 002670 004737 003262 1$: JSR PC,CHANGE ;;CHECK/CHANGE THE RH/RM ADDRESS
34 002674 104401 071552 TYPE ,ENTERA ;;ENTER DRIVE ADDRESS
35 002700 104412 RDOCT ;;GET THE ADDRESS
36 002702 012637 001224 MOV (SP)+,PORTA ;;STORE THE ADDRESS
37 002706 023727 001224 000007 CMP PORTA,#7 ;;SEE IF ADDRESS TOO LARGE
38 002714 101403 BLOS 2$ ;;BR IF NOT
39 002716 104401 071601 TYPE ,ADRERR ;;TYPE ADDRESS ERROR MESSAGE
40 002722 000762 BR 1$ ;;TRY AGAIN
41 002724 013737 001224 001226 2$: MOV PORTA,PORTB ;;GENERATE THE PORT B ADDRESS
42 002732 005237 001226 INC PORTB ;;INCREMENT THE ADDRESS
43 002736 042737 000016 001226 BIC #16,PORTB ;;LEAVE BIT 0
44 002744 013746 001224 MOV PORTA,-(SP) ;;PUT PORT A ADDRESS ON THE STACK
45 002750 042716 177771 BIC #^C6,(SP) ;;SAVE BITS 1 & 2
46 002754 052637 001226 BIS (SP)+,PORTB ;;SET BITS 1 & 2 IN PORT B ADDRESS
47 002760 104401 071623 TYPE ,PORTAIS ;;'PORT A ADDRESS IS '
48 002764 013746 001224 MOV PORTA,-(SP) ;;SAVE PORTA FOR TYPEOUT
;;TYPE PORT A ADDRESS
;;GO TYPE--OCTAL ASCII
002770 104403 TYPOS 1 ;;TYPE 1 DIGIT(S)
002772 001 .BYTE 1 ;;SUPPRESS LEADING ZEROS
002773 000 .BYTE 0 ;;'PORT B ADDRESS IS '
49 002774 104401 071650 TYPE ,PORTBIS ;;SAVE PORTB FOR TYPEOUT
50 003000 013746 001226 MOV PORTB,-(SP) ;;TYPE PORT B ADDRESS

```



```

003004 104403
003006 001
003007 000
51 003010 104401 001207
52 003014 013737 001224 001230
53 003022 062737 000006 001230
54 003030 042737 177770 001230
55 003036 013701 001224
56 003042 116137 101052 001236
59 003050 005037 001262
003054 005037 001264
003060 005037 001270
003064 005037 001272
60 003070 004737 066124
61 003074 000137 003110
62 003100 104401 071675
63 003104 000000 3$:
64 003106 000776 BR 3$
65
66 ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
67
68 003110 000005 EXEC: RESET ;CLEAR EVERYTHING
69 003112 005037 177776 CLR PS ;CLEAR THE PROCESSOR STATUS WORD
70 003116 104401 001207 TYPE , $CRLF ;CR-LF
71 003122 013700 001304 MOV $RMADR,RO ;RH/RM ADDRESS FOR INDEXING
72 003126 012706 001100 MOV #STACK,SP ;LOAD STACK POINTER
73 003132 004737 066124 JSR PC,CKCLK ;START THE CLOCK
74 003136 000240 NOP ;RETURN IF NO CLOCK
75 003140 004737 070074 JSR PC,$TKINT ;INITIALIZE THE KEYBOARD
76 003144 005037 001300 CLR KYBCTL ;CLEAR SINGLE TEST INDICATOR
77 003150 005037 001100 CLR $PASS ;CLEAR THE PASS COUNT
78 003154 112737 000001 001115 MOV #1,$ERMAX ;SET ERROR MAX TO 1
79 003162 012737 003162 001106 MOV #,$LPADR ;INITIAL SETTING FOR LOOP ADDRESS
80 003170 012737 003170 001110 MOV #,$LPERR ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
81 003176 104401 071733 1$: TYPE ,TESTNO ;ASK FOR TEST NUMBER
82 003202 104412 RDOCT ;GET THE NUMBER
83 003204 012601 MOV (SP)+,R1 ;PUT ENTRY INTO R1
84 003206 001002 BNE 2$ ;BR IF NOT ZERO
85 003210 000137 003376 JMP TST1 ;ENTER ZERO - PERFORM ALL TESTS
86 003214 020137 101062 2$: CMP R1,MAXTN ;SEE IF NUMBER GREATER THAN MAXIMUM
87 003220 003403 BLE 3$ ;BR IF LESS OR EQUAL
88 003222 104401 071753 TYPE ,BADNO ;BAD ENTRY
89 003226 000763 BR 1$ ;TRY AGAIN
90 003230 005301 3$: DEC R1 ;DECREMENT ENTRY
91 003232 006301 ASL R1 ;SHIFT IT LEFT
92 003234 016137 100736 003260 MOV TSTADR(R1),4$ ;GET THE TEST ADDRESS
93 003242 005237 001300 INC KYBCTL ;SET SINGLE TEST INDICATOR
94 003246 012737 000001 001104 MOV #1,$ICNT ;PRESET ITERATION COUNT
95 003254 000177 000000 JMP @4$ ;GO TO THE SELECTED TEST
96 003260 000000 4$: .WORD 0 ;TEST ADDRESS GOES HERE
97
98 ;CHANGE THE RH/RM UNIBUS ADDRESS USED BY THE PROGRAM
99
100 003262 005737 001302 CHANGE: TST CHGADR ;CHANGE THE ADDRESS ?
101 003266 001421 BEQ 3$ ;BR IF NOT
102 003270 005037 001302 CLR CHGADR ;CLEAR THE INDICATOR
103 003274 104401 072012 1$: TYPE ,ADDRIS ;TYPE OUT WHAT THE PRESENT ADDRESS IS

```



```

104 003300 013746 001304      MOV      $RMADR,-(SP)      ;PUT THE ADDRESS ON THE STACK
105 003304 104402              TYPOC                    ;TYPE THE ACTUAL ADDRESS
106 003306 104401 001207      TYPE      ,%CR LF        ;CR-LF
107 003312 104401 072047      TYPE      ,NTRH11        ;ASK FOR NEW ADDRESS
108 003316 104412              RDOCT
109 003320 005716              TST      (SP)            ;0 OR 'CR' ENTERED ?
110 003322 001402              BEQ      2$              ;BR IF EITHER ENTERED (NO ADDRESS CHANGE)
111 003324 011637 001304      MOV      (SP), $RMADR    ;NEW RH/RM ADDRESS
112 003330 005726              TST      (SP)+          ;CORRECT THE STACK POINTER
113 003332 012737 003352 000004 2$:      MOV      #4$, @#4        ;LOAD TRAP ADDRESS
114 003340 013700 001304      MOV      $RMADR, R0      ;GET RH/RM ADDRESS
115 003344 005760 000002      TST      RMC(R0)        ;RESPONDS AT THAT ADDRESS ?
116 003350 000404              BR       5$              ;BR IF YES
117 003352              4$:
118 003354 104056              EMT      56
119 003360 062706 000004      ADD      #4, SP          ;RESET THE STACK POINTER
120 003362 000745              BR       1$              ;GET ADDRESS AGAIN
121 003362 012737 000006 000004 5$:      MOV      #6, @#4        ;RESTORE THE VECTOR
122 003370 000207              RTS      PC              ;RETURN
136
137 003372 013700 001304      TST1AA: MOV      $RMADR, R0 ;;RESTORE R0 AFTER END OF PASS
138
139
    
```

\*\*\*\*\*  
 \*TEST 1 NEUTRAL ACCESS TEST  
 \*\*\*\*\*

\*VERIFY THAT THE DRIVE IS ACCESSIBLE TO BOTH PORTS

\* A. SELECT DRIVE, VERIFY THAT THE DRIVE IS PRESENT, THAT  
 \* DRIVE IS A DUAL PORT RM05, RM03 OR RM02 AND THAT THE DRIVE  
 \* IS ONLINE (RMDS HAS 'MOL', 'PGM', 'DPR', & 'DRY' BITS SET),  
 \* AND THE THE DRIVE SERIAL NUMBER READ THROUGH BOTH PORTS IS  
 \* THE SAME.

\* B. THE TEST IS REPEATED THROUGH BOTH PORTS.

\*\*\*\*\*

```

003376
003376 005737 001300      TST1:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
003402 001406              BEQ      2$              ;BR IF NOT
003404 100002              BPL      1$              ;BR IF JUST ENTERED TEST
003406 000137 003110      JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
003412 012737 177777 001300 1$:      MOV      #-1, KYBCTL     ;SET SINGLE TEST INDICATOR
003420 012737 003434 001106 2$:      MOV      #TEST1, $LPADR  ;SETUP SCOPE LOOP ADDRESS
003426 012737 003434 001110      MOV      #TEST1, $LPERR  ;SETUP ERROR LOOP ADDRESS
003434
003434 112737 000001 001102      TEST1: MOVB     #1, $STNM      ;MOVE #1 TO TEST NUMBER
003442 012706 001100      MOV      #STACK, SP     ;LOAD THE STACK POINTER
003446 012737 000001 001176      MOV      #1, $TIMES     ;;DO 1 ITERATION
140
141 003454 012760 000040 000010      MOV      #CLR, RMCS2(R0) ;INITIALIZE THE MASSBUS
142
143      ;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B
144
152 003462 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
003470 013737 001224 001240      MOV      PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003476 005760 000012      TST      RMDS(R0)      ;SEE IF DRIVE (PORT A) PRESENT
    
```



```
003502 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
003506 016037 000010 001126 MOV RMCS2(RO), $BDDAT ;GET CONTENTS OF RMCS2
003514 012737 000010 001122 MOV #RMCS2, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003522 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
003526 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
003532 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
003540 042737 167777 001164 BIC #^CNED, $TMP0 ;SAVE SPECIFIED BITS
003546 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
003554 001414 BEQ 64$ ;BR IF OK
003556 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
003564 042737 010000 001174 BIC #NED, $TMP4 ;CLEAR THE MASKED BITS
003572 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
003600 104037 EMT 37
003602 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
003606 000240 64$: NOP
003610 005737 001250 TST CKERR ;WAS 'NED' SET ?
003614 001403 BEQ .+10 ;BR IF NOT
003616 012760 000040 000010 MOV #CLR, RMCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
003624 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B
003632 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003640 005760 000012 TST RMD5(RO) ;SEE IF DRIVE (PORT B) PRESENT
003644 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
003650 016037 000010 001126 MOV RMCS2(RO), $BDDAT ;GET CONTENTS OF RMCS2
003656 012737 000010 001122 MOV #RMCS2, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003664 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
003670 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
003674 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
003702 042737 167777 001164 BIC #^CNED, $TMP0 ;SAVE SPECIFIED BITS
003710 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
003716 001414 BEQ 66$ ;BR IF OK
003720 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
003726 042737 010000 001174 BIC #NED, $TMP4 ;CLEAR THE MASKED BITS
003734 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
003742 104037 EMT 37
003744 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
003750 000240 66$: NOP
003752 005737 001250 TST CKERR ;WAS 'NED' SET ?
003756 001403 BEQ .+10 ;BR IF NOT
003760 012760 000040 000010 MOV #CLR, RMCS2(RO) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
153
154 ;CONFIRM THAT DRIVE IS AN RM05, RM03 OR RM02 AND IS DUAL PORTED
155
159 003766 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
003774 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004002 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
004006 016037 000026 001126 MOV RMDT(RO), $BDDAT ;GET CONTENTS OF RMDT
004014 012737 000026 001122 MOV #RMDT, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004022 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
004026 012737 024027 001124 MOV #024027, $GDDAT ;WHAT REGISTER SHOULD BE
004034 022737 024024 001126 CMP #024024, $BDDAT ;DUAL PORT RM03 ?
004042 001413 BEQ 68$ ;YES !!
004044 022737 024025 001126 CMP #024025, $BDDAT ;DUAL PORT RM02 ?
004052 001407 BEQ 68$ ;YES !!
004054 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER OK ?
004062 001403 BEQ 68$ ;BR IF OK
004064 104001 EMT 1
004066 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
```



```

004072 000240          68$: NOP
004074 113760 001226 000010 MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
004102 013737 001226 001240 MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004110 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
004114 016037 000026 001126 MOV   RMDT(RO), $BDDAT ;GET CONTENTS OF RMDT
004122 012737 000026 001122 MOV   #RMDT, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004130 060037 001122          ADD   RO, $BDADR ;ADD RH/RM BASE ADDRESS
004134 012737 024027 001124 MOV   #024027, $GDDAT ;WHAT REGISTER SHOULD BE
004142 022737 024024 001126 CMP   #024024, $BDDAT ;DUAL PORT RM03 ?
004150 001413          BEQ   70$ ;YES !!
004152 022737 024025 001126 CMP   #024025, $BDDAT ;DUAL PORT RM02 ?
004160 001407          BEQ   70$ ;YES !!
004162 023737 001124 001126 CMP   $GDDAT, $BDDAT ;IS THE REGISTER OK ?
004170 001403          BEQ   70$ ;BR IF OK
004172 104001          EMT   1
004174 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
004200 000240          70$: NOP

;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL

160
161
162
167 004202 113760 001224 000010 MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
004210 013737 001224 001240 MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004216 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
004222 016037 000012 001126 MOV   RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
004230 012737 000012 001122 MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004236 060037 001122          ADD   RO, $BDADR ;ADD RH/RM BASE ADDRESS
004242 012737 001000 001124 MOV   #PGM, $GDDAT ;WHAT REGISTER SHOULD BE
004250 013737 0 1126 001164 MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
004256 042737 176777 001164 BIC   #^CPGM, $TMP0 ;SAVE SPECIFIED BITS
004264 023737 001124 001164 CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
004272 001414          BEQ   72$ ;BR IF OK
004274 013737 001126 001174 MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
004302 042737 001000 001174 BIC   #PGM, $TMP4 ;CLEAR THE MASKED BITS
004310 053737 001174 001124 BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004316 104045          EMT   45
004320 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
004324 000240          72$: NOP
004326 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
004332 016037 000012 001126 MOV   RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
004340 012737 000012 001122 MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004346 060037 001122          ADD   RO, $BDADR ;ADD RH/RM BASE ADDRESS
004352 012737 010600 001124 MOV   #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
004360 013737 001126 001164 MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
004366 042737 167177 001164 BIC   #^C10600, $TMP0 ;SAVE SPECIFIED BITS
004374 023737 001124 001164 CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
004402 001414          BEQ   74$ ;BR IF OK
004404 013737 001126 001174 MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
004412 042737 010600 001174 BIC   #10600, $TMP4 ;CLEAR THE MASKED BITS
004420 053737 001174 001124 BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004426 104002          EMT   2
004430 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
004434 000240          74$: NOP
004436 113760 001226 000010 MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
004444 013737 001226 001240 MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004452 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
004456 016037 000012 001126 MOV   RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
004464 012737 000012 001122 MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```



```

004472 060037 001122          ADD    RO,$BDADR          ;ADD RH/RM BASE ADDRESS
004476 012737 001000 001124  MOV    #PGM,$GDDAT       ;WHAT REGISTER SHOULD BE
004504 013737 001126 001164  MOV    $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
004512 042737 176777 001164  BIC    #^CPGM,$TMP0      ;SAVE SPECIFIED BITS
004520 023737 001124 001164  CMP    $GDDAT,$TMP0      ;COMPARE THE BITS
004526 001414          BEQ    76$                ;BR IF OK
004530 013737 001126 001174  MOV    $BDDAT,$TMP4      ;COPY 'BAD DATA'
004536 042737 001000 001174  BIC    #PGM,$TMP4        ;CLEAR THE MASKED BITS
004544 053737 001174 001124  BIS    $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
004552 104045          EMT    45
004554 005137 001250          COM    CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
004560 000240          76$: NOP
004562 005037 001250          CLR    CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
004566 016037 000012 001126  MOV    RMDS(RO),$BDDAT    ;GET CONTENTS OF RMCS
004574 012737 000012 001122  MOV    #RMDS,$BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004602 060037 001122          ADD    RO,$BDADR          ;ADD RH/RM BASE ADDRESS
004606 012737 010600 001124  MOV    #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
004614 013737 001126 001164  MOV    $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
004622 042737 167177 001164  BIC    #^C10600,$TMP0    ;SAVE SPECIFIED BITS
004630 023737 001124 001164  CMP    $GDDAT,$TMP0      ;COMPARE THE BITS
004636 001414          BEQ    78$                ;BR IF OK
004640 013737 001126 001174  MOV    $BDDAT,$TMP4      ;COPY 'BAD DATA'
004646 042737 010600 001174  BIC    #10600,$TMP4      ;CLEAR THE MASKED BITS
004654 053737 001174 001124  BIS    $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
004662 104002          EMT    2
004664 005137 001250          COM    CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
004670 000240          78$: NOP

168
169
170          ;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME
171 004672 113760 001224 000010  MOVB   PORTA,RMCS2(RO)   ;SELECT PORT A
172 004700 016037 000030 001124  MOV    RMSN(RO),$GDDAT   ;STORE THE PORT A SERIAL NUMBER
173 004706 113760 001226 000010  MOVB   PORTB,RMCS2(RO)   ;SELECT PORT B
174 004714 016037 000030 001126  MOV    RMSN(RO),$BDDAT   ;STORE THE PORT B SERIAL NUMBER
175 004722 023737 001124 001126  CMP    $GDDAT,$BDDAT    ;ARE THEY THE SAME ?
176 004730 001406          BEQ    1$                ;BR IF THEY ARE
177 004732 104003          EMT    3
178 004734 032777 100000 174176  BIT    #SW15,@SWR        ;HALT ON ERROR ?
179 004742 001001          BNE    1$                ;BR IF SET - PROGRAM HAS ALREADY HALTED
180 004744 000000          HALT
181 004746 000004          1$: SCOPE              ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
182
200          ;LOOP ?
201

```

```

:*****
:*TEST 2          PORT 'A' SEIZE/TIMEOUT TEST
:*
:*VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT
:*IT CAN BE RELEASED BY THE ONE SECOND TIMER.
:*
:*  A.  WRITE 0'S INTO RMDA THROUGH PORT 'A'; VERIFY THAT THE DRIVE
:*      HAS BEEN SEIZED.
:*
:*  B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'B';
:*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
:*
:*  C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
:*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE

```

;\* VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO  
 :\* NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS > 500 MS.  
 :\*

\*\*\*\*\*

```

004750
004750 005737 001300
004754 001406
004756 100002
004760 000137 003110
004764 012737 177777 001300 1$:
004772 012737 005006 001106 2$:
005000 012737 005006 001110
005006
005006 112737 000002 001102
005014 012706 001100
005020 012737 000002 001176
202
270 005026 012737 000240 177776
005034 005037 001262
005040 005037 001264
005044 005037 001266

TST2:
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
MOV #TEST2,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST2,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST2:
MOVB #2,$TSTNM ;MOVE #2 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #2.,$TIMES ;;DO 2. ITERATIONS

MOV #<5*32.>,@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
CLR TIMEA ;CLEAR TIMEOUT VALUE FOR PORT A
CLR TIMEAP ;CLEAR UPPER TIMEOUT TOLERANCE
CLR TIMEAM ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

005050 005037 001256
005054 012737 003720 001260
CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
MOV #2000.,WATCH ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT A

005062 113760 001224 000010
005070 013737 001224 001242
005076 005060 000006
005102 113760 001226 000010
005110 013737 001226 001240
005116 013737 001226 001244
005124 016037 000012 001126
005132 010037 001122
005136 062737 000012 001122
005144 005037 001124
005150 023737 001124 001126
005156 001403
005160 104004
005162 000137 006314
005166
64$:
005166 113760 001224 000010
005174 013737 001224 001240
005202 016037 000012 001126
005210 042737 020001 001126
005216 012737 011600 001124
005224 013737 001124 001166
005232 005137 001166
005236 013737 001126 001164
005244 043737 001166 001164
005252 023737 001124 001164
005260 001401
005262 104005
005264 000240

MOVB PORTA, RMCS2(R0) ;SELECT PORT A
MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDA(R0) ;WRITE RMDA
MOVB PORTB, RMCS2(R0) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RMDA(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO, $BADDR ;RH/RM BASE ADDRESS
ADD #RMDA, $BADDR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
BEQ 64$ ;BR IF IT IS
EMT 4
JMP 5$ ;BYPASS REST OF THE SUBTEST

MOVB PORTA, RMCS2(R0) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMDA(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
MOV #MOL!PGM!DPR!DRY, $GDDAT ;EXPECTED STATUS
MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
BEQ 65$ ;BR IF THEY ARE
EMT 5
NOP
65$:
  
```



;READ THE DRIVE REGISTERS THROUGH PORT B AND STORE THEM ON THE STACK

```

005266 113760 001226 000010      MOV  PORTB, RMCS2(R0) ;SELECT PORT B
005274 013737 001226 001240      MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005302 016046 000046              MOV  RMEC2(R0), -(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
005306 016046 000044              MOV  RMEC1(R0), -(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
005312 016046 000030              MOV  RMSN(R0), -(SP) ;STORE REGISTER RMSN, PORT B, FOR CHECK
005316 016046 000034              MOV  RMDC(R0), -(SP) ;STORE REGISTER RMDC, PORT B, FOR CHECK
005322 016046 000032              MOV  RMOF(R0), -(SP) ;STORE REGISTER RMOF, PORT B, FOR CHECK
005326 016046 000042              MOV  RMER2(R0), -(SP) ;STORE REGISTER RMER2, PORT B, FOR CHECK
005332 016046 000020              MOV  RMLA(R0), -(SP) ;STORE REGISTER RMLA, PORT B, FOR CHECK
005336 016046 000026              MOV  RMDT(R0), -(SP) ;STORE REGISTER RMDT, PORT B, FOR CHECK
005342 016046 000006              MOV  RMDA(R0), -(SP) ;STORE REGISTER RMDA, PORT B, FOR CHECK
005346 016046 000024              MOV  RMMR1(R0), -(SP) ;STORE REGISTER RMMR1, PORT B, FOR CHECK
005352 016046 000014              MOV  RMER1(R0), -(SP) ;STORE REGISTER RMER1, PORT B, FOR CHECK
  
```

;WAIT FOR PORT A TO TIMEOUT

```

005356 005760 000012      1$: TST  RMDS(R0) ;WAIT FOR THE DRIVE TO TIMEOUT
005362 001006              BNE  2$ ;BR WHEN TIMEOUT OCCURS
005364 005737 001260      TST  WATCH ;CHECK WATCH
005370 001372              BNE  1$ ;BR IF NOT ZERO
005372 104036              EMT  36
005374 000137 006000      JMP  4$ ;BYPASS TIMEOUT TIME CHECK
005400 012737 000340 177776 2$: MOV  #<7*32.>, @#PS ;SET PRIORITY TO 7 TO STOP CLOCK
005406 013737 001256 001262      MOV  TIME, TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
005414 004537 066320      JSR  R5, TOLER ;CALCULATE THE TOLERANCE
005420 001262              .WORD TIMEA ;TIMEOUT VALUE FOR PORT A
005422 012637 001264      MOV  (SP)+, TIMEAP ;+25% TOLERANCE
005426 012637 001266      MOV  (SP)+, TIMEAM ;-25% TOLERANCE
  
```

;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS

```

005432 023727 001256 000764      CMP  TIME, #500. ;WAS MEASURED TIME AT LEAST 500 MS?
005440 103001              BHIS 3$ ;BR IF IT WAS
005442 104055              EMT  55
  
```

;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT A TIMED OUT

```

005444 012737 000240 177776 3$: MOV  #<5*32.>, @#PS ;RESTORE PRIORITY TO 5
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

005452 005037 001254              CLR  RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
005456 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
005464 060037 001122              ADD  R0, $BDADR ;ADD THE I/O BASE ADDRESS
005470 012737 011600 001124      MOV  #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
005476 113760 001224 000010      MOV  PORTA, RMCS2(R0) ;SELECT PORT A.
005504 016037 000012 001170      MOV  RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
005512 042737 024001 001170      BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
005520 013737 001170 001164      MOV  $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
005526 042737 100100 001164      BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005534 113760 001226 000010      MOV  PORTB, RMCS2(R0) ;SELECT PORT B.
005542 016037 000012 001172      MOV  RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
005550 042737 024001 001172      BIC  #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
005556 013737 001172 001166      MOV  $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
  
```

```

005564 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005572 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
005600 001006 BNE 66$ ;BR IF NOT
005602 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
005606 001037 BNE 68$ ;BR IF NOT
005610 104046 EMT 46
005612 000137 005776 JMP 70$ ;BYPASS THE REST OF THE CHECKS
005616 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
005624 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005632 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
005640 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
005644 001414 BEQ 67$ ;BR IF ZERO
005646 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005654 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
005662 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
005670 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
005674 001004 BNE 68$ ;BR IF NOT
005676 012737 177777 001254 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
005704 104022 EMT 22
005706 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
005714 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
005722 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
005730 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
005736 001401 BEQ 69$ ;BR IF OK FROM PORT A.
005740 104007 EMT 7
005742 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
005750 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
005756 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
005764 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
005772 001401 BEQ 70$ ;BR IF OK
005774 104007 EMT 7
005776 000240 70$: NOP
    
```

;CHECK THE REGISTERS STORED THROUGH PORT B. ALL REGISTERS SHOULD BE ZERO.  
 ;THE REGISTERS ARE STORED ON THE STACK.

```

006000 013737 001226 001240 4$: MOV PORTB,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
006006 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER1
006012 062737 000014 001122 ADD #RMER1,$BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
006020 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER1
006024 001401 BEQ .+4 ;CONTENTS ZERO ?
006026 104006 EMT 6
006030 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMMR1
006034 062737 000024 001122 ADD #RMMR1,$BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
006042 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
006046 001401 BEQ .+4 ;CONTENTS ZERO ?
006050 104006 EMT 6
006052 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
006056 062737 000006 001122 ADD #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
006064 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
006070 001401 BEQ .+4 ;CONTENTS ZERO ?
006072 104006 EMT 6
006074 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
006100 062737 000026 001122 ADD #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
006106 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
006112 001401 BEQ .+4 ;CONTENTS ZERO ?
006114 104006 EMT 6
    
```



```

006116 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMLA
006122 062737 000020 001122  ADD    #RMLA,$BDADR  ;ADDRESS OF RMLA FOR TIMEOUT
006130 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMLA
006134 001401              BEQ    .+4            ;CONTENTS ZERO ?
006136 104006              EMT    6
006140 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMER2
006144 062737 000042 001122  ADD    #RMER2,$BDADR ;ADDRESS OF RMER2 FOR TIMEOUT
006152 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMER2
006156 001401              BEQ    .+4            ;CONTENTS ZERO ?
006160 104006              EMT    6
006162 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMOF
006166 062737 000032 001122  ADD    #RMOF,$BDADR  ;ADDRESS OF RMOF FOR TIMEOUT
006174 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMOF
006200 001401              BEQ    .+4            ;CONTENTS ZERO ?
006202 104006              EMT    6
006204 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMDC
006210 062737 000034 001122  ADD    #RMDC,$BDADR  ;ADDRESS OF RMDC FOR TIMEOUT
006216 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMDC
006222 001401              BEQ    .+4            ;CONTENTS ZERO ?
006224 104006              EMT    6
006226 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMSN
006232 062737 000030 001122  ADD    #RMSN,$BDADR  ;ADDRESS OF RMSN FOR TIMEOUT
006240 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMSN
006244 001401              BEQ    .+4            ;CONTENTS ZERO ?
006246 104006              EMT    6
006250 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC1
006254 062737 000044 001122  ADD    #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TIMEOUT
006262 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMEC1
006266 001401              BEQ    .+4            ;CONTENTS ZERO ?
006270 104006              EMT    6
006272 010037 001122      MOV    RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC2
006276 062737 000046 001122  ADD    #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TIMEOUT
006304 012637 001126      MOV    (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMEC2
006310 001401              BEQ    .+4            ;CONTENTS ZERO ?
006312 104006              EMT    6
006314 000004              5$:   SCOPE          ;LOOP ?
  
```

271  
289  
290

```

*****
*TEST 3      PORT 'B' SEIZE/TIMEOUT TEST
*
*VERIFY THAT THE DRIVE CAN BE SEIZED BY WRITING A REMOTE REGISTER AND THAT
*IT CAN BE RELEASED BY THE ONE SECOND TIMER.
*
*  A.  WRITE 0'S INTO RMDA THROUGH PORT 'B'; VERIFY THAT THE DRIVE
*      HAS BEEN SEIZED.
*
*  B.  READ EACH DRIVE REGISTER, EXCEPT RMCS1, THROUGH PORT 'A';
*      VERIFY THAT 0'S ARE READ FROM EACH REGISTER.
*
*  C.  WAIT FOR THE PORT TIMEOUT TO RELEASE THE DRIVE.
*      MEASURE THE DURATION OF THE TIMEOUT ONE SHOT AND SAVE THE
*      VALUE FOR LATER USE. VERIFY THAT TIMEOUT RETURNED THE DRIVE TO
*      NEUTRAL. ALSO VERIFY THAT THE DURATION OF THE ONE SHOT IS >500 MS.
*****
TST3:
  
```

006316

```

006316 005737 001300      TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
006322 001406      BEQ      2$          ;BR IF NOT
006324 100002      BPL      1$          ;BR IF JUST ENTERED TEST
006326 000137 003110      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
006332 012737 177777 001300 1$:      MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
006340 012737 006354 001106 2$:      MOV      #TEST3,$LPADR ;SETUP SCOPE LOOP ADDRESS
006346 012737 006354 001110      MOV      #TEST3,$LPERR ;SETUP ERROR LOOP ADDRESS
006354      TEST3:
006354 112737 000003 001102      MOVB     #3,$TSTNM    ;MOVE #3 TO TEST NUMBER
006362 012706 001100      MOV      #STACK,SP   ;LOAD THE STACK POINTER
006366 012737 000002 001176      MOV      #2.,$TIMES   ;;DO 2. ITERATIONS

291
292 006374 012737 000240 177776      MOV      #<5*32.>,@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
006402 005037 001270      CLR      TIMEB        ;CLEAR TIMEOUT VALUE FOR PORT B
006406 005037 001272      CLR      TIMEBP       ;CLEAR UPPER TIMEOUT TOLERANCE
006412 005037 001274      CLR      TIMEBM       ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

006416 005037 001256      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
006422 012737 003720 001260      MOV      #2000.,WATCH ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT B

006430 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
006436 013737 001226 001242      MOV      PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
006444 005060 000006      CLR      RMDA(R0)     ;WRITE RMDA
006450 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
006456 013737 001224 001240      MOV      PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006464 013737 001224 001244      MOV      PORTA, OPPRT  ;'OPPOSITE' PORT ADDRESS
006472 016037 000012 001126      MOV      RMD5(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
006500 010037 001122      MOV      R0, $BDADR    ;RH/RM BASE ADDRESS
006504 062737 000012 001122      ADD      #RMD5, $BDADR ;GENERATE BAD REGISTER ADDRESS
006512 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
006516 023737 001124 001126      CMP      $GDDAT, $BDDAT ;IS THE REGISTER ZERO
006524 001403      BEQ      64$          ;BR IF IT IS
006526 104004      EMT      4
006530 000137 007662      JMP      5$            ;BYPASS REST OF THE SUBTEST
006534      64$:
006534 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
006542 013737 001226 001240      MOV      PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
006550 016037 000012 001126      MOV      RMD5(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
006556 042737 020001 001126      BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
006564 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY, $GDDAT ;EXPECTED STATUS
006572 013737 001124 001166      MOV      $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
006600 005137 001166      COM      $TMP1         ;COMPLEMENT THE EXPECTED STATUS
006604 013737 001126 001164      MOV      $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
006612 043737 001166 001164      BIC      $TMP1, $TMP0  ;CLEAR UNWANTED BITS
006620 023737 001124 001164      CMP      $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
006626 001401      BEQ      65$          ;BR IF THEY ARE
006630 104005      EMT      5
006632 000240      65$:      NOP

;READ THE DRIVE REGISTERS THROUGH PORT A AND STORE THEM ON THE STACK

006634 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
006642 013737 001224 001240      MOV      PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```



```

006650 016046 000046      MOV      RMEC2(RO),-(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
006654 016046 000044      MOV      RMEC1(RO),-(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
006660 016046 000030      MOV      RMSN(RO),-(SP) ;STORE REGISTER RMSN, PORT A, FOR CHECK
006664 016046 000034      MOV      RMDC(RO),-(SP) ;STORE REGISTER RMDC, PORT A, FOR CHECK
006670 016046 000032      MOV      RMOF(RO),-(SP) ;STORE REGISTER RMOF, PORT A, FOR CHECK
006674 016046 000042      MOV      RMER2(RO),-(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
006700 016046 000020      MOV      RMLA(RO),-(SP) ;STORE REGISTER RMLA, PORT A, FOR CHECK
006704 016046 000026      MOV      RMDT(RO),-(SP) ;STORE REGISTER RMDT, PORT A, FOR CHECK
006710 016046 000006      MOV      RMDA(RO),-(SP) ;STORE REGISTER RMDA, PORT A, FOR CHECK
006714 016046 000024      MOV      RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
006720 016046 000014      MOV      RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK
    
```

;WAIT FOR PORT B TO TIMEOUT

```

006724 005760 000012      1$:     TST      RMDS(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
006730 001006 ;BR WHEN TIMEOUT OCCURS
006732 005737 001260      TST      WATCH ;CHECK WATCH
006736 001372 ;BR IF NOT ZERO
006740 104036      EMT      36
006742 000137 007346      JMP      4$ ;BYPASS TIMEOUT TIME CHECK
006746 012737 000340 177776 2$:     MOV      #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
006754 013737 001256 001270      MOV      TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
006762 004537 066320      JSR      R5,TOLER ;CALCULATE THE TOLERANCE
006766 001270 ;TIMEOUT VALUE FOR PORT B
006770 012637 001272      MOV      (SP)+,TIMEBP ;+25% TOLERANCE
006774 012637 001274      MOV      (SP)+,TIMEBM ;-25% TOLERANCE
    
```

;VERIFY THAT THE TIMEOUT ONE-SHOT IS AT LEAST 500 MS

```

007000 023727 001256 000764      CMP      TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
007006 103001 ;BR IF IT WAS
007010 104055      EMT      55
    
```

;VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AFTER PORT B TIMED OUT

```

007012 012737 000240 177776 3$:     MOV      #<5*32.>,@#PS ;RESTORE PRIORITY TO 5
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

007020 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
007024 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
007032 060037 001122      ADD      RO,$BDADR ;ADD THE I/O BASE ADDRESS
007036 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
007044 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
007052 016037 000012 001170      MOV      RMDS(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
007060 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
007066 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
007074 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007102 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
007110 016037 000012 001172      MOV      RMDS(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
007116 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
007124 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
007132 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007140 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
007146 001006 ;BR IF NOT
007150 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
007154 001037 ;BR IF NOT
    
```

```

007156 104046          EMT      46
007160 000137 007344    JMP      70$
007164 013737 001170 001126 66$:    MOV     $TMP2,$BDDAT ;BYPASS THE REST OF THE CHECKS
007172 013737 001226 001240    MOV     PORTB,PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
007200 113760 001226 000010    MOVB   PORTB,RMCS2(RO) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007206 005737 001164          TST     $TMP0 ;SELECT PORT B.
007212 001414          BEQ     67$ ;SEE IF STATUS EQ 0 FROM PORT A.
007214 013737 001224 001240    MOV     PORTA,PTNBR ;BR IF ZERO
007222 013737 001172 001126    MOV     $TMP3,$BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007230 113760 001224 000010    MOVB   PORTA,RMCS2(RO) ;'BAD DATA' FOR ERROR TYPE OUT
007236 005737 001166          TST     $TMP1 ;SELECT PORT A.
007242 001004          BNE     68$ ;SEE IF STATUS EQ ZERO FROM PORT B.
007244 012737 177777 001254 67$:    MOV     #-1,RELERR ;BR IF NOT
007252 104022          EMT     22 ;SET 'RELEASE ERROR' INDICATOR
007254 013737 001170 001126 68$:    MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
007262 013737 001224 001240    MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
007270 042737 100100 001126    BIC    #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007276 023737 001124 001126    CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
007304 001401          BEQ     69$ ;BR IF OK FROM PORT A.
007306 104007          EMT     7
007310 013737 001172 001126 69$:    MOV     $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
007316 013737 001226 001240    MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
007324 042737 100100 001126    BIC    #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007332 023737 001124 001126    CMP    $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
007340 001401          BEQ     70$ ;BR IF OK
007342 104007          EMT     7
007344 000240          70$:   NOP
    
```

;CHECK THE REGISTERS STORED THROUGH PORT A. ALL REGISTERS SHOULD BE ZERO.  
 ;THE REGISTERS ARE STORED ON THE STACK.

```

007346 013737 001224 001240 4$:    MOV     PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007354 010037 001122          MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER1
007360 062737 000014 001122    ADD    #RMER1,$BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
007366 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER1
007372 001401          BEQ     .+4 ;CONTENTS ZERO ?
007374 104006          EMT     6
007376 010037 001122          MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RMMR1
007402 062737 000024 001122    ADD    #RMMR1,$BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
007410 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
007414 001401          BEQ     .+4 ;CONTENTS ZERO ?
007416 104006          EMT     6
007420 010037 001122          MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
007424 062737 000006 001122    ADD    #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
007432 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
007436 001401          BEQ     .+4 ;CONTENTS ZERO ?
007440 104006          EMT     6
007442 010037 001122          MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
007446 062737 000026 001122    ADD    #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
007454 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
007460 001401          BEQ     .+4 ;CONTENTS ZERO ?
007462 104006          EMT     6
007464 010037 001122          MOV     RO,$BDADR ;BASE ADDRESS FOR REGISTER RMLA
007470 062737 000020 001122    ADD    #RMLA,$BDADR ;ADDRESS OF RMLA FOR TYPEOUT
007476 012637 001126          MOV     (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMLA
007502 001401          BEQ     .+4 ;CONTENTS ZERO ?
007504 104006          EMT     6
    
```



```

007506 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMER2
007512 062737 000042 001122  ADD      #RMER2,$BDADR  ;ADDRESS OF RMER2 FOR TYPEOUT
007520 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMER2
007524 001401              BEQ      .+4           ;CONTENTS ZERO ?
007526 104006              EMT      6
007530 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMOF
007534 062737 000032 001122  ADD      #RMOF,$BDADR  ;ADDRESS OF RMOF FOR TYPEOUT
007542 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMOF
007546 001401              BEQ      .+4           ;CONTENTS ZERO ?
007550 104006              EMT      6
007552 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMDC
007556 062737 000034 001122  ADD      #RMDC,$BDADR  ;ADDRESS OF RMDC FOR TYPEOUT
007564 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMDC
007570 001401              BEQ      .+4           ;CONTENTS ZERO ?
007572 104006              EMT      6
007574 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMSN
007600 062737 000030 001122  ADD      #RMSN,$BDADR  ;ADDRESS OF RMSN FOR TYPEOUT
007606 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMSN
007612 001401              BEQ      .+4           ;CONTENTS ZERO ?
007614 104006              EMT      6
007616 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC1
007622 062737 000044 001122  ADD      #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TYPEOUT
007630 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMEC1
007634 001401              BEQ      .+4           ;CONTENTS ZERO ?
007636 104006              EMT      6
007640 010037 001122      MOV      RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC2
007644 062737 000046 001122  ADD      #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TYPEOUT
007652 012637 001126      MOV      (SP)+,$BDDAT  ;CHECK THE STORED CONTENTS OF RMEC2
007656 001401              BEQ      .+4           ;CONTENTS ZERO ?
007660 104006              EMT      6
007662 000004      5$: SCOPE ;LOOP ?
  
```

293  
307  
308

```

*****
*TEST 4      PORT 'A' SEIZE/RELEASE TEST
*
*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDs.
*
*  B.  SET VOLUME VALID AND CLEAR ANY ERROR
*
*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
*      DRIVE.
*****
  
```

```

007664 005737 001300      TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
007664 001406              BEQ      2$          ;BR IF NOT
007672 100002              BPL      1$          ;BR IF JUST ENTERED TEST
007674 000137 003110      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
007700 012737 177777 001300  1$: MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
007706 012737 007722 001106  2$: MOV      #TEST4,$LPADR ;SETUP SCOPE LOOP ADDRESS
007714 012737 007722 001110      MOV      #TEST4,$LPERR ;SETUP ERROR LOOP ADDRESS
007722
007722 112737 000004 001102  TEST4: MOV      #4,$STNM   ;MOVE #4 TO TEST NUMBER
  
```

309  
342

```

007730 012706 001100          MOV    #STACK,SP      ;LOAD THE STACK POINTER
007734 012737 000031 001176  MOV    #25.,$TIMES   ;:DO 25. ITERATIONS

                                ;:START THE TIMER

007742 005037 001256          CLR    TIME          ;CLEAR THE ELAPSED TIME COUNTER
007746 012737 003720 001260  MOV    #2000.,WATCH ;SET WATCH TO 2000. MS

                                ;:SEIZE THE DRIVE AND SET VOLUME VALID
                                ;:SEIZE THE DRIVE THROUGH PORT A

007754 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
007762 013737 001224 001242  MOV    PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
007770 005060 000012          CLR    RMDS(R0)      ;WRITE RMDS
007774 013737 001226 001244  MOV    PORTB, OPPRT  ;'OPPOSITE' PORT ADDRESS
010002 012760 000021 000000  MOV    #21, RMCS1(R0) ;SET VOLUME VALID
010010 005037 001250          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
010014 016037 000012 001126  MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
010022 012737 000012 001122  MOV    #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010030 060037 001122          ADD    R0, $BDADR   ;ADD RH/RM BASE ADDRESS
010034 012737 000100 001124  MOV    #VV, $GDDAT  ;WHAT REGISTER SHOULD BE
010042 013737 001126 001164  MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010050 042737 177677 001164  BIC    #^CVV, $TMP0 ;SAVE SPECIFIED BITS
010056 023737 001124 001164  CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
010064 001414          BEQ    66$          ;BR IF OK
010066 013737 001126 001174  MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
010074 042737 000100 001174  BIC    #VV, $TMP4   ;CLEAR THE MASKED BITS
010102 053737 001174 001124  BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010110 104013          EMT    13
010112 005137 001250          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
010116 000240          NOP
010120 012760 000040 000010  MOV    #CLR, RMCS2(R0) ;CLEAR DRIVE

                                ;:RELEASE THE DRIVE FROM PORT A

010126 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
010134 013737 001224 001240  MOV    PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010142 012760 000013 000000  MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A

                                ;:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

010150 005037 001254          CLR    RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
010154 012737 000012 001122  MOV    #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
010162 060037 001122          ADD    R0, $BDADR   ;ADD THE I/O BASE ADDRESS
010166 012737 011600 001124  MOV    #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
010174 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
010202 016037 000012 001170  MOV    RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
010210 042737 024001 001170  BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
010216 013737 001170 001164  MOV    $TMP2, $TMP0  ;COPY IT INTO '$TMP0'
010224 042737 100100 001164  BIC    #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010232 113760 001226 000010  MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
010240 016037 000012 001172  MOV    RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
010246 042737 024001 001172  BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
010254 013737 001172 001166  MOV    $TMP3, $TMP1  ;COPY IT INTO '$TMP1'
010262 042737 100100 001166  BIC    #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
  
```



```

010270 023737 001164 001166      CMP      $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
010276 001006                      BNE      68$              ;BR IF NOT
010300 005737 001164                      TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
010304 001037                      BNE      70$              ;BR IF NOT
010306 104046                      EMT      46
010310 000137 010474                      JMP      72$              ;BYPASS THE REST OF THE CHECKS
010314 013737 001170 001126 68$:    MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
010322 013737 001226 001240          MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010330 113760 001226 000010          MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B.
010336 005737 001164                      TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
010342 001414                      BEQ      69$              ;BR IF ZERO
010344 013737 001224 001240          MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010352 013737 001172 001126          MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
010360 113760 001224 000010          MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
010366 005737 001166                      TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
010372 001004                      BNE      70$              ;BR IF NOT
010374 012737 177777 001254 69$:    MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
010402 104022                      EMT      22
010404 013737 001170 001126 70$:    MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
010412 013737 001224 001240          MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
010420 042737 100100 001126          BIC      #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
010426 023737 001124 001126          CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
010434 001401                      BEQ      71$              ;BR IF OK FROM PORT A.
010436 104007                      EMT      7
010440 013737 001172 001126 71$:    MOV      $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
010446 013737 001226 001240          MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
010454 042737 100100 001126          BIC      #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
010462 023737 001124 001126          CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
010470 001401                      BEQ      72$              ;BR IF OK
010472 104007                      EMT      7
010474 000240                      NOP
010476 005737 001254 72$:    TST      RELERR          ;DID DRIVE RETURN TO NEUTRAL ?
010502 001402                      BEQ      +6              ;BR IF IN NEUTRAL
010504 000137 010760                      JMP      1$              ;GO WAIT FOR DRIVE TO TIMEOUT
010510 113760 001224 000010          MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A
010516 013737 001224 001240          MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010524 005037 001250                      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010530 016037 000012 001126          MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
010536 012737 000012 001126          MOV      #RMDS,$BADDR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010544 060037 001122                      ADD      RO,$BADDR       ;ADD RH/RM BASE ADDRESS
010550 005037 001124                      CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
010554 013737 001126 001164          MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
010562 042737 077777 001164          BIC      #^CATA,$TMP0    ;SAVE SPECIFIED BITS
010570 023737 001124 001164          CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
010576 001414                      BEQ      73$              ;BR IF OK
010600 013737 001126 001174          MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
010606 042737 100000 001174          BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
010614 053737 001174 001124          BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
010622 104017                      EMT      17
010624 005137 001250 73$:    COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
010630 000240                      NOP
010632 113760 001226 000010          MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
010640 013737 001226 001240          MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010646 005037 001250                      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
010652 016037 000012 001126          MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
010660 012737 000012 001126          MOV      #RMDS,$BADDR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010666 060037 001122                      ADD      RO,$BADDR       ;ADD RH/RM BASE ADDRESS
    
```



```

010672 005037 001124          CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
010676 013737 001126 001164  MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
010704 042737 077777 001164  BIC      #^CATA,$STMP0 ;SAVE SPECIFIED BITS
010712 023737 001124 001164  CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
010720 001414          BEQ      75$         ;BR IF OK
010722 013737 001126 001174  MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
010730 042737 100000 001174  BIC      #ATA,$STMP4   ;CLEAR THE MASKED BITS
010736 053737 001174 001124  BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010744 104017          EMT      17
010746 005137 001250          COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
010752 000240          75$:  NOP
010754 000137 011012          JMP      2$         ;GO CHECK FOR LOOP ON ERROR
  
```

;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT  
 ;TO RELEASE THE DRIVE

```

010760          1$:
010760 113760 001226 000010  MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
010766 013737 001226 001240  MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010774 005760 000012          TST     RMDS(R0)       ;WAIT FOR TIMEOUT TO RELEASE DRIVE
011000 001004          BNE     2$           ;BR WHEN DRIVE RELEASED
011002 005737 001260          TST     WATCH        ;CHECK THE WATCH
011006 001364          BNE     1$           ;BR IF NOT ZERO
011010 104036          EMT     36
011012 000004          2$:  SCOPE          ;LOOP ?
  
```

343  
 357  
 358

```

*****
*TEST 5      PORT 'B' SEIZE/RELEASE TEST
*****
*TEST THE OPERATION OF THE RELEASE COMMAND, DRIVE SEIZED
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
*
*  B.  SET VOLUME VALID AND CLEAR ANY ERROR
*
*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.  VERIFY THAT THE DRIVE
*      RETURNED TO NEUTRAL, AND THAT NO ERRORS ARE INDICATED BY THE
*      DRIVE.
*****
  
```

```

011014          TST5:
011014 005737 001300          TST     KYBCTL       ;PERFORMING ONLY SINGLE TEST ?
011020 001406          BEQ     2$           ;BR IF NOT
011022 100002          BPL     1$           ;BR IF JUST ENTERED TEST
011024 000137 003110          JMP     EXEC         ;RETURN & GET NEXT TEST NUMBER
011030 012737 177777 001300  1$:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
011036 012737 011052 001106  2$:  MOV     #TEST5,$LPADR ;SETUP SCOPE LOOP ADDRESS
011044 012737 011052 001110  MOV     #TEST5,$LPERR  ;SETUP ERROR LOOP ADDRESS
011052          TST5:
011052 112737 000005 001102  MOVB    #5,$STSTNM   ;MOVE #5 TO TEST NUMBER
011060 012706 001100          MOV     #STACK,SP   ;LOAD THE STACK POINTER
011064 012737 000031 001176  MOV     #25,$TIMES   ;;DO 25. ITERATIONS
  
```

359  
 360

;START THE TIMER



```

011072 005037 001256          CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
011076 012737 003720 001260  MOV      #2000.,WATCH ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE AND SET VOLUME VALID
;SEIZE THE DRIVE THROUGH PORT B

011104 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
011112 013737 001226 001242  MOV      PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
011120 005060 000012          CLR      RMDS(R0)       ;WRITE RMDS
011124 013737 001224 001244  MOV      PORTA, OPPRT   ;'OPPOSITE' PORT ADDRESS
011132 012760 000021 000000  MOV      #21, RMCS1(R0) ;SET VOLUME VALID
011140 005037 001250          CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
011144 016037 000012 001126  MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
011152 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011160 060037 001122          ADD      R0, $BDADR    ;ADD RH/RM BASE ADDRESS
011164 012737 000100 001124  MOV      #VV, $GDDAT   ;WHAT REGISTER SHOULD BE
011172 013737 001126 001164  MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011200 042737 177677 001164  BIC      #^CVV, $TMP0  ;SAVE SPECIFIED BITS
011206 023737 001124 001164  CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
011214 001414          BEQ      66$         ;BR IF OK
011216 013737 001126 001174  MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
011224 042737 000100 001174  BIC      #VV, $TMP4   ;CLEAR THE MASKED BITS
011232 053737 001174 001124  BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
011240 104013          EMT      13
011242 005137 001250          COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
011246 000240          NOP
011250 012760 000040 000010  MOV      #CLR, RMCS2(R0) ;CLEAR DRIVE

;RELEASE THE DRIVE FROM PORT B

011256 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
011264 013737 001226 001240  MOV      PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011272 012760 000013 000000  MOV      #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

011300 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
011304 012737 000012 001122  MOV      #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
011312 060037 001122          ADD      R0, $BDADR    ;ADD THE I/O BASE ADDRESS
011316 012737 011600 001124  MOV      #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
011324 113760 001224 000010  MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
011332 016037 000012 001170  MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
011340 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
011346 013737 001170 001164  MOV      $TMP2, $TMP0  ;COPY IT INTO '$TMP0'
011354 042737 100100 001164  BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011362 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B.
011370 016037 000012 001172  MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
011376 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
011404 013737 001172 001166  MOV      $TMP3, $TMP1  ;COPY IT INTO '$TMP1'
011412 042737 100100 001166  BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011420 023737 001164 001166  CMP      $TMP0, $TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
011426 001006          BNE      68$         ;BR IF NOT
011430 005737 001164          TST      $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
011434 001037          BNE      70$         ;BR IF NOT
011436 104046          EMT      46
011440 000137 011624          JMP      72$         ;BYPASS THE REST OF THE CHECKS
    
```

```

011444 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
011452 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011460 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
011466 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
011472 001414 BEQ 69$ ;BR IF ZERO
011474 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011502 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
011510 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
011516 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
011522 001004 BNE 70$ ;BR IF NOT
011524 012737 177777 001254 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
011532 104022 EMT 22
011534 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
011542 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
011550 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011556 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
011564 001401 BEQ 71$ ;BR IF OK FROM PORT A.
011566 104007 EMT 7
011570 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
011576 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
011604 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011612 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
011620 001401 BEQ 72$ ;BR IF OK
011622 104007 EMT 7
011624 000240 72$: NOP
011626 005737 001254 TST RELERR ;DID DRIVE RETURN TO NEUTRAL ?
011632 001402 BEQ .+6 ;BR IF IN NEUTRAL
011634 000137 012110 JMP 1$ ;GO WAIT FOR DRIVE TO TIMEOUT
011640 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B
011646 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011654 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
011660 016037 000012 001126 MOV RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
011666 012737 000012 001122 MOV #RMDS,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011674 060037 001122 ADD RO,$BADDR ;ADD RH/RM BASE ADDRESS
011700 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
011704 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011712 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
011720 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
011726 001414 BEQ 73$ ;BR IF OK
011730 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
011736 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
011744 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
011752 104017 EMT 17
011754 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
011760 000240 73$: NOP
011762 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
011770 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011776 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
012002 016037 000012 001126 MOV RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
012010 012737 000012 001122 MOV #RMDS,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
012016 060037 001122 ADD RO,$BADDR ;ADD RH/RM BASE ADDRESS
012022 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
012026 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
012034 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
012042 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
012050 001414 BEQ 75$ ;BR IF OK
012052 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
    
```



```

012060 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
012066 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
012074 104017                      EMT      17
012076 005137 001250              COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
012102 000240              75$:  NOP
012104 000137 012142              JMP      2$            ;GO CHECK FOR LOOP ON ERROR
  
```

;IF RELEASE COMMAND DIDN'T RELEASE THE DRIVE, WAIT FOR THE PORT TIMEOUT  
 ;TO RELEASE THE DRIVE

```

012110 113760 001224 000010      1$:  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
012110 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012116 013737 001224 001240      TST   RMDS(R0)        ;WAIT FOR TIMEOUT TO RELEASE DRIVE
012124 005760 000012              BNE   2$              ;BR WHEN DRIVE RELEASED
012130 001004                      TST   WATCH           ;CHECK THE WATCH
012132 005737 001260              BNE   1$              ;BR IF NOT ZERO
012136 001364                      EMT   36
012140 104036                      2$:  SCOPE            ;LOOP ?
012142 000004
  
```

365  
 374  
 375

\*\*\*\*\*  
 ;\*TEST 6 PORT 'A' NEUTRAL/RELEASE TEST  
 ;\*

;\*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL  
 ;\*

;\* A. ISSUE A RELEASE COMMAND THROUGH PORT 'A' WITH THE DRIVE IN  
 ;\* NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.  
 ;\*

```

012144 005737 001300              TST6:  TST    KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
012144 001406                      BEQ    2$              ;BR IF NOT
012152 100002                      BPL    1$              ;BR IF JUST ENTERED TEST
012154 000137 003110              JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
012160 012737 177777 001300      1$:  MOV    #-1, KYBCTL   ;SET SINGLE TEST INDICATOR
012166 012737 012202 001106      2$:  MOV    #TEST6,$LPADR ;SETUP SCOPE LOOP ADDRESS
012174 012737 012202 001110      MOV    #TEST6,$LPERR   ;SETUP ERPOR LOOP ADDRESS
012202                                TEST6:  MOVB   #6,$STNM        ;MOVE #6 TO TEST NUMBER
012202 112737 000006 001102      MOV    #STACK, SP      ;LOAD THE STACK POINTER
012210 012706 001100              MOV    #25., $TIMES    ;;DO 25. ITERATIONS
012214 012737 000031 001176
  
```

376  
 387

```

012222 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
012230 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012236 013737 001224 001242      MOV    PORTA, SEIZPT   ;ADDR OF PORT WHICH WILL ISSUE RELEASE
  
```

```

012244 012760 000013 000000      ;ISSUE A RELEASE COMMAND
MOV    #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

012252 005037 001254              CLR    RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
012256 012737 000012 001122      MOV    #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
012264 060037 001122              ADD    RO,$BDADR      ;ADD THE I/O BASE ADDRESS
012270 012737 011700 001124      MOV    #MQL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
012276 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
  
```

```

012304 016037 000012 001170      MOV      RMDS(R0), $TMP2      ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012312 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2      ;CLEAR DONT CARES
012320 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
012326 042737 100100 001164      BIC      #ATA!VV, $TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012334 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
012342 016037 000012 001172      MOV      RMDS(R0), $TMP3      ;GET THE DRIVE STATUS REGISTER FROM PORT B.
012350 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3      ;CLEAR DONT CARES
012356 013737 001172 001166      MOV      $TMP3, $TMP1      ;COPY IT INTO '$TMP1'
012364 042737 100100 001166      BIC      #ATA!VV, $TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
012372 023737 001164 001166      CMP      $TMP0, $TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
012400 001006 001164 001166      BNE      64$      ;BR IF NOT
012402 005737 001164 001166      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
012406 001045 001164 001166      BNE      66$      ;BR IF NOT
012410 104046 001164 001166      EMT      46
012412 000137 012612 001166      JMP      68$      ;BYPASS THE REST OF THE CHECKS
012416 013737 001170 001126 64$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012424 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012432 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
012440 005737 001164 001166      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
012444 001414 001164 001166      BEQ      65$      ;BR IF ZERO
012446 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012454 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
012462 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)      ;SELECT PORT A.
012470 005737 001166 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
012474 001012 001166 001166      BNE      66$      ;BR IF NOT
012476 012737 177777 001254 65$:  MOV      #-1, RELERR      ;SET 'RELEASE ERROR' INDICATOR
012504 012760 000011 000000      MOV      #11, RMCS1(R0)      ;CLEAR THE DRIVE
012512 012760 000013 000000      MOV      #13, RMCS1(R0)      ;RELEASE THE DRIVE
012520 104030 001166 001166      EMT      30
012522 013737 001170 001126 66$:  MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
012530 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT NUMBER
012536 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
012544 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;ALL BITS OK ?
012552 001401 001124 001126      BEQ      67$      ;BR IF OK FROM PORT A.
012554 104007 001166 001166      EMT      7
012556 013737 001172 001126 67$:  MOV      $TMP3, $BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
012564 013737 001226 001240      MOV      PORTB, PTNBR      ;CHANGE PORT NUMBER
012572 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
012600 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;SEE IF READ OK FROM PORT B.
012606 001401 001124 001126      BEQ      68$      ;BR IF OK
012610 104007 001124 001126      EMT      7
012612 000240 001124 001126 68$:  NOP
012614 000004 001124 001126      SCOPE      ;LOOP ?
    
```

388  
397  
398

```

*****
:*TEST 7      PORT 'B' NEUTRAL/RELEASE TEST
:*
:*TEST OPERATION OF RELEASE COMMAND, DRIVE IN NEUTRAL
:*
:*  A.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' WITH THE DRIVE IN
:*      NEUTRAL; VERIFY THAT THE DRIVE REMAINS IN NEUTRAL.
:*
*****
    
```

```

012616 005737 001300 001300      TST7:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
012616 001406 001300 001300      BEQ      2$      ;BR IF NOT
012622 001406 001300 001300
    
```



```

012624 100002          BPL      1$          ;BR IF JUST ENTERED TEST
012626 000137 003110  JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
012632 012737 177777 001300 1$:      MOV      #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
012640 012737 012654 001106 2$:      MOV      #TEST7,$LPADR ;SETUP SCOPE LOOP ADDRESS
012646 012737 012654 001110      MOV      #TEST7,$LPERR ;SETUP ERROR LOOP ADDRESS
012654          TEST7:
012654 112737 000007 001102      MOVB     #7,$STSTM      ;MOVE #7 TO TEST NUMBER
012662 012706 001100      MOV      #STACK,SP     ;LOAD THE STACK POINTER
012666 012737 000031 001176      MOV      #25.,$TIMES    ;DO 25. ITERATIONS
399
400 012674 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
012702 013737 001226 001240      MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012710 013737 001226 001242      MOV      PORTB, SEIZPT  ;ADDR OF PORT WHICH WILL ISSUE RELEASE

;ISSUE A RELEASE COMMAND
012716 012760 000013 000000      MOV      #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

012724 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
012730 012737 000012 001122      MOV      #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
012736 060037 001122          ADD      R0,$BDADR     ;ADD THE I/O BASE ADDRESS
012742 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
012750 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
012756 016037 000012 001170      MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012764 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
012772 013737 001170 001164      MOV      $TMP2, $TMP0   ;COPY IT INTO '$TMP0'
013000 042737 100100 001164      BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013006 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B.
013014 016037 000012 001172      MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
013022 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
013030 013737 001172 001166      MOV      $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
013036 042737 100100 001166      BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013044 023737 001164 001166      CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
013052 001006          BNE      64$          ;BR IF NOT
013054 005737 001164          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
013060 001045          BNE      66$          ;BR IF NOT
013062 104046          EMT      46
013064 000137 013264          JMP      68$          ;BYPASS THE REST OF THE CHECKS
013070 013737 001170 001126 64$:    MOV      $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
013076 013737 001226 001240      MOV      PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013104 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B.
013112 005737 001164          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
013116 001414          BEQ      65$          ;BR IF ZERO
013120 013737 001224 001240      MOV      PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013126 013737 001172 001126      MOV      $TMP3, $BDDAT  ;'BAD DATA' FOR ERROR TYPE OUT
013134 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
013142 005737 001166          TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
013146 001012          BNE      66$          ;BR IF NOT
013150 012737 177777 001254 65$:    MOV      #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
013156 012760 000011 000000      MOV      #11, RMCS1(R0) ;CLEAR THE DRIVE
013164 012760 000013 000000      MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
013172 104030          EMT      30
013174 013737 001170 001126 66$:    MOV      $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
013202 013737 001224 001240      MOV      PORTA, PTNBR   ;CHANGE PORT NUMBER
013210 042737 100000 001126      BIC      #ATA, $BDDAT   ;DON'T CHECK THE ATTN BIT
013216 023737 001124 001126      CMP      $GDDAT, $BDDAT ;ALL BITS OK ?

```

```

013224 001401          BEQ      67$          ;BR IF OK FROM PORT A.
013226 104007          EMT      7
013230 013737 001172 001126 67$:  MOV      $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
013236 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
013244 042737 100000 001126      BIC      #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
013252 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
013260 001401          BEQ      68$          ;BR IF OK
013262 104007          EMT      7
013264 000240          68$:  NOP
013266 000004          SCOPE          ;LOOP ?
    
```

401  
420  
421

```

:*****
:TEST 10          PORT 'A' RELEASE INTERFERENCE TEST
:
:VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
:IS SEIZED BY THE OTHER PORT.
:
:A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
:
:B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'.
:
:C. VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'B'.
:
:D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE SWITCHED
:TO PORT 'A'.
:
:E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED
:TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*****
    
```

```

013270
013270 005737 001300      TST10:  TST      KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
013274 001406          BEQ      2$          ;BR IF NOT
013276 100002          BPL      1$          ;BR IF JUST ENTERED TEST
013300 000137 003110      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
013304 012737 177777 001300 1$:  MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
013312 012737 013326 001106 2$:  MOV      #TEST10,$LPADR ;SETUP SCOPE LOOP ADDRESS
013320 012737 013326 001110      MOV      #TEST10,$LPERR ;SETUP ERROR LOOP ADDRESS
013326
013326 112737 000010 001102 TEST10: MOVB     #10,$STNM      ;MOVE #10 TO TEST NUMBER
013334 012706 001100      MOV      #STACK,SP   ;LOAD THE STACK POINTER
013340 012737 000031 001176      MOV      #25.,$TIMES ;DO 25. ITERATIONS
    
```

422  
446

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

013346 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT #A
013354 005060 000012          CLR      RMDS(R0)       ;SEIZE THE DRIVE
013360 012760 000011 000000      MOV      #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
013366 012760 000013 000000      MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
013374 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT #B
013402 005060 000012          CLR      RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
013406 012760 000011 000000      MOV      #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
013414 012760 000013 000000      MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
    
```

;SEIZE THE DRIVE THROUGH PORT B



```

013422 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
013430 013737 001226 001242      MOV   PORTB, SEIZPT  ;STORE SEIZING PORT'S ADDRESS
013436 005060 000012                CLR   RMDS(RO)       ;WRITE RMDS
013442 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
013450 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013456 013737 001224 001244      MOV   PORTA, OPPRT  ;'OPPOSITE' PORT ADDRESS
013464 016037 000012 001126      MOV   RMDS(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
013472 010037 001122                MOV   RO, $BDADR    ;RH/RM BASE ADDRESS
013476 062737 000012 001122      ADD   #RMDS, $BDADR  ;GENERATE BAD REGISTER ADDRESS
013504 005037 001124                CLR   $GDDAT        ;REGISTER SHOULD BE ZERO
013510 023737 001124 001126      CMP   $GDDAT, $BDDAT ;IS THE REGISTER ZERO
013516 001403                BEQ   64$           ;BR IF IT IS
013520 104004                EMT
013522 000137 014532                JMP   1$           ;BYPASS REST OF THE SUBTEST
013526                                64$:
013526 113760 001226 000010      MOVB  PORTB, RMCS2(RO) ;SELECT PORT B
013534 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013542 016037 000012 001126      MOV   RMDS(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
013550 042737 020001 001126      BIC   #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
013556 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
013564 013737 001124 001166      MOV   $GDDAT, $TMP1  ;USE GOOD DATA AS A MASK
013572 005137 001166                COM   $TMP1         ;COMPLEMENT THE EXPECTED STATUS
013576 013737 001126 001164      MOV   $BDDAT, $TMP0  ;SAVE THE ACTUAL STATUS
013604 043737 001166 001164      BIC   $TMP1, $TMP0  ;CLEAR UNWANTED BITS
013612 023737 001124 001164      CMP   $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
013620 001401                BEQ   65$           ;BR IF THEY ARE
013622 104005                EMT
013624 000240                                65$:
                                NOP

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

013626 113760 001224 000010      MOVB  PORTA, RMCS2(RO) ;SELECT PORT A
013634 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013642 012760 000013 000000      MOV   #13, RMCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

013650 005037 001250                CLR   CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
013654 016037 000012 001126      MOV   RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
013662 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
013670 060037 001122                ADD   RO, $BDADR    ;ADD RH/RM BASE ADDRESS
013674 005037 001124                CLR   $GDDAT        ;WHAT REGISTER SHOULD BE
013700 023737 001124 001126      CMP   $GDDAT, $BDDAT ;IS THE REGISTER OK ?
013706 001403                BEQ   66$           ;BR IF OK
013710 104010                EMT
013712 005137 001250                COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
013716 016037 000000 001126      66$: MOV   RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RMCS1
013724 012737 000000 001122      MOV   #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
013732 060037 001122                ADD   RO, $BDADR    ;ADDRESS BASE
013736 032737 020000 001126      BIT   #MCPE, $BDDAT ;IS 'MCPE' SET ?
013744 001404                BEQ   67$           ;BR IF NOT
013746 104011                EMT
013750 012760 040000 000000      67$: MOV   #TRE, RMCS1(RO) ;CLEAR 'MCPE'
013756 000240                NOP
013760 005737 001250                TST   CKERR         ;WAS RMDS NON ZERO ?
013764 001402                BEQ   .+6          ;CONTENTS OF RMDS SEEN BY PORT A
  
```

013766 000137 014532 JMP 1\$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

;RELEASE THE DRIVE FROM PORT B

013772 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B  
 014000 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 014006 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B

014014 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR  
 014020 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT  
 014026 012737 000012 001122 MOV #RMDS,\$BDADR ;REGISTER ADDRESS INCREMENT  
 014034 060037 001122 ADD RO,\$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT  
 014040 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A  
 014046 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 014054 016037 000012 001164 MOV RMDS(RO), \$TMP0 ;READ STATUS REGISTER FROM PORT A  
 014062 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B  
 014070 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 014076 016037 000012 001126 MOV RMDS(RO), \$BDDAT ;DRIVE STATUS FROM PORT B  
 014104 001404 BEQ 68\$ ;BR IF STATUS FROM PORT B ZERO  
 014106 005737 001164 TST \$TMP0 ;IS STATUS FROM PORT A ZERO ?  
 014112 001401 BEQ 68\$ ;BR IF ZERO  
 014114 104031 EMT 31  
 014116 013737 001164 001126 68\$: MOV \$TMP0,\$BDDAT ;CHECK STATUS FROM PORT A  
 014124 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT  
 014132 023737 001124 001126 CMP \$GDDAT,\$BDDAT ;COMPARE WITH CONSTANT  
 014140 001401 BEQ 69\$ ;BR IF OK  
 014142 104027 EMT 27  
 014144 000240 69\$: NOP

;RELEASE THE DRIVE FROM PORT A

014146 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A  
 014154 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 014162 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

014170 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR  
 014174 012737 000012 001122 MOV #RMDS,\$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT  
 014202 060037 001122 ADD RO,\$BDADR ;ADD THE I/O BASE ADDRESS  
 014206 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT  
 014214 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.  
 014222 016037 000012 001170 MOV RMDS(RO), \$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.  
 014230 042737 024001 001170 BIC #PIP!WRL!OM,\$TMP2 ;CLEAR DONT CARES  
 014236 013737 001170 001164 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'  
 014244 042737 100100 001164 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY  
 014252 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.  
 014260 016037 000012 001172 MOV RMDS(RO), \$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.  
 014266 042737 024001 001172 BIC #PIP!WRL!OM,\$TMP3 ;CLEAR DONT CARES  
 014274 013737 001172 001166 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'  
 014302 042737 100100 001166 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY  
 014310 023737 001164 001166 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?  
 014316 001006 BNE 70\$ ;BR IF NOT  
 014320 005737 001164 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?  
 014324 001045 BNE 72\$ ;BR IF NOT



```

014326 104046          EMT      46
014330 000137 014530  JMP      74$
014334 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;BYPASS THE REST OF THE CHECKS
014342 013737 001226 001240  MOV     PORTB,PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014350 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014356 005737 001164          TST     $TMP0 ;SELECT PORT B.
014362 001414          BEQ     71$ ;SEE IF STATUS EQ 0 FROM PORT A.
014364 013737 001224 001240  MOV     PORTA,PTNBR ;BR IF ZERO
014372 013737 001172 001126  MOV     $TMP3,$BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014400 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;'BAD DATA' FOR ERROR TYPE OUT
014406 005737 001166          TST     $TMP1 ;SELECT PORT A.
014412 001012          BNE     72$ ;SEE IF STATUS EQ ZERO FROM PORT B.
014414 012737 177777 001254 71$:  MOV     #-1,RELERR ;BR IF NOT
014422 012760 000011 000000  MOV     #11,RMCS1(RO) ;SET 'RELEASE ERROR' INDICATOR
014430 012760 000013 000000  MOV     #13,RMCS1(RO) ;CLEAR THE DRIVE
014436 104026          EMT      26 ;RELEASE THE DRIVE
014440 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
014446 013737 001224 001240  MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
014454 042737 100000 001126  BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
014462 023737 001124 001126  CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
014470 001401          BEQ     73$ ;BR IF OK FROM PORT A.
014472 104007          EMT      7
014474 013737 001172 001126 73$:  MOV     $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
014502 013737 001226 001240  MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
014510 042737 100000 001126  BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
014516 023737 001124 001126  CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
014524 001401          BEQ     74$ ;BR IF OK
014526 104007          EMT      7
014530 000240          NOP
014532 000004          1$:  SCOPE ;LOOP ?
  
```

447  
466  
467

```

*****
*TEST 11      PORT 'B' RELEASE INTERFERENCE TEST
*
*VERIFY THAT A COMMAND ISSUED BY ONE PORT IS NOT RECOGNIZED IF THE DRIVE
*  IS SEIZED BY THE OTHER PORT.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*
*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'.
*
*  C.  VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT 'A'.
*
*  D.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE SWITCHED
*      TO PORT 'B'.
*
*  E.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
  
```

```

014534          TST11:
014534 005737 001300  TST     KYBCTL ;PERFORMING ONLY SINGLE TEST ?
014540 001406          BEQ     2$ ;BR IF NOT
014542 100002          BPL     1$ ;BR IF JUST ENTERED TEST
014544 000137 003110  JMP     EXEC ;RETURN & GET NEXT TEST NUMBER
014550 012737 177777 001300 1$:  MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
  
```

468  
469

```

014556 012737 014572 001106 2$: MOV #TEST11,$LPADR ;SETUP SCOPE LOOP ADDRESS
014564 012737 014572 001110 MOV #TEST11,$LPERR ;SETUP ERROR LOOP ADDRESS
014572 TEST11:
014572 112737 000011 001102 MOV #11,$STNM ;MOVE #11 TO TEST NUMBER
014600 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
014604 012737 000031 001176 MOV #25.,$TIMES ;;DO 25. ITERATIONS
  
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

C14612 113760 001224 000010 MOV #PORTA, RMCS2(RO) ;SELECT PORT #A
014620 005060 000012 CLR RMDS(RO) ;SEIZE THE DRIVE
014624 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
014632 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
014640 113760 001226 000010 MOV #PORTB, RMCS2(RO) ;SELECT PORT #B
014646 005060 000012 CLR RMDS(RO) ;SEIZE THE DRIVE THROUGH PORT 'B'
014652 012760 000011 000000 MOV #11, RMCS1(RO) ;ISSUE DRIVE CLEAR
014660 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

014666 113760 001224 000010 MOV #PORTA, RMCS2(RO) ;SELECT PORT A
014674 013737 001224 001242 MOV #PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
014702 005060 000012 CLR RMDS(RO) ;WRITE RMDS
014706 113760 001226 000010 MOV #PORTB, RMCS2(RO) ;SELECT PORT B
014714 013737 001226 001240 MOV #PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014722 013737 001226 001244 MOV #PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
014730 016037 000012 001126 MOV #RMDS(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
014736 010037 001122 MOV #RO, $BDADR ;RH/RM BASE ADDRESS
014742 062737 000012 001122 ADD #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
014750 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
014754 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
014762 001403 BEQ 64$ ;BR IF IT IS
014764 104004 EMT 4
014766 000137 015776 JMP 1$ ;BYPASS REST OF THE SUBTEST
014772
  
```

64\$:

```

014772 113760 001224 000010 MOV #PORTA, RMCS2(RO) ;SELECT PORT A
015000 013737 001224 001240 MOV #PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015006 016037 000012 001126 MOV #RMDS(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
015014 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
015022 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
015030 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
015036 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
015042 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
015050 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
015056 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
015064 001401 BEQ 65$ ;BR IF THEY ARE
015066 104005 EMT 5
015070 000240
  
```

65\$:

```

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B
015072 113760 001226 000010 MOV #PORTB, RMCS2(RO) ;SELECT PORT B
015100 013737 001226 001240 MOV #PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015106 012760 000013 000000 MOV #13, RMCS1(RO) ;ISSUE A RELEASE COMMAND THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A



```

015114 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
015120 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
015126 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015134 060037 001122          ADD      RO, $BDADR     ;ADD RH/RM BASE ADDRESS
015140 005037 001124          CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
015144 023737 001124 001126  CMP      $GDDAT, $BDDAT ;IS THE REGISTER OK ?
015152 001403          BEQ      66$           ;BR IF OK
015154 104010          EMT      10
015156 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
015162 016037 000000 001126 66$:  MOV      RMCS1(RO), $BDDAT ;GET THE CONTENTS OF RMCS1
015170 012737 000000 001122  MOV      #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
015176 060037 001122          ADD      RO, $BDADR     ;ADDRESS BASE
015202 032737 020000 001126  BIT      #MCPE, $BDDAT  ;IS 'MCPE' SET ?
015210 001404          BEQ      67$           ;BR IF NOT
015212 104011          EMT      11
015214 012760 040000 000000  MOV      #TRE, RMCS1(RO) ;CLEAR 'MCPE'
015222 000240          NOP
015224 005737 001250          TST      CKERR          ;WAS RMDS NON ZERO ?
015230 001402          BEQ      .+6           ;CONTENTS OF RMDS SEEN BY PORT B
015232 000137 015776          JMP      1$            ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
  
```

;RELEASE THE DRIVE FROM PORT A

```

015236 113760 001224 000010  MOVB     PORTA, RMCS2(RO) ;SELECT PORT A
015244 013737 001224 001240  MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015252 012760 000013 000000  MOV      #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A

```

015260 005037 001254          CLR      RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
015264 012737 111700 001124  MOV      #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
015272 012737 000012 001122  MOV      #RMDS, $BDADR  ;REGISTER ADDRESS INCREMENT
015300 060037 001122          ADD      RO, $BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
015304 113760 001226 000010  MOVB     PORTB, RMCS2(RO) ;SELECT PORT B
015312 013737 001226 001240  MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015320 016037 000012 001164  MOV      RMDS(RO), $TMPO ;READ STATUS REGISTER FROM PORT B
015326 113760 001224 000010  MOVB     PORTA, RMCS2(RO) ;SELECT PORT A
015334 013737 001224 001240  MOV      PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015342 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;DRIVE STATUS FROM PORT A
015350 001404          BEQ      68$           ;BR IF STATUS FROM PORT A ZERO
015352 005737 001164          TST      $TMPO         ;IS STATUS FROM PORT B ZERO ?
015356 001401          BEQ      68$           ;BR IF ZERO
015360 104031          EMT      31
015362 013737 001164 001126 68$:  MOV      $TMPO, $BDDAT  ;CHECK STATUS FROM PORT B
015370 013737 001226 001240  MOV      PORTB, PTNBR   ;CHANGE PORT ADDRESS FOR TYPEOUT
015376 023737 001124 001126  CMP      $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
015404 001401          BEQ      69$           ;BR IF OK
015406 104027          EMT      27
015410 000240          NOP
  
```

;RELEASE THE DRIVE FROM PORT B

```

015412 113760 001226 000010  MOVB     PORTB, RMCS2(RO) ;SELECT PORT B
015420 013737 001226 001240  MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015426 012760 000013 000000  MOV      #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B
  
```



:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

015434 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
015440 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
015446 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
015452 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
015460 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
015466 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
015474 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
015502 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
015510 042737 100100 001164  BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015516 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
015524 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
015532 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
015540 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
015546 042737 100100 001166  BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015554 023737 001164 001166  CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
015562 001006          BNE      70$        ;BR IF NOT
015564 005737 001164          TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
015570 001045          BNE      72$        ;BR IF NOT
015572 104046          EMT      46
015574 000137 015774          JMP      74$        ;BYPASS THE REST OF THE CHECKS
015600 013737 001170 001126 70$:  MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
015606 013737 001226 001240  MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015614 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
015622 005737 001164          TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
015626 001414          BEQ      71$        ;BR IF ZERO
015630 013737 001224 001240  MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015636 013737 001172 001126  MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
015644 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
015652 005737 001166          TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
015656 001012          BNE      72$        ;BR IF NOT
015660 012737 177777 001254 71$:  MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
015666 012760 000011 000000  MOV      #11, RMCS1(RO) ;CLEAR THE DRIVE
015674 012760 000013 000000  MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
015702 104026          EMT      26
015704 013737 001170 001126 72$:  MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
015712 013737 001224 001240  MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
015720 042737 100000 001126  BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
015726 023737 001124 001126  CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
015734 001401          BEQ      73$        ;BR IF OK FROM PORT A.
015736 104007          EMT      7
015740 013737 001172 001126 73$:  MOV      $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
015746 013737 001226 001240  MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
015754 042737 100000 001126  BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
015762 023737 001124 001126  CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
015770 001401          BEQ      74$        ;BR IF OK
015772 104007          EMT      7
015774 000240          NOP
015776 000004          1$:  SCOPE          ;LOOP ?
  
```

470  
490  
491

```

*****
*TEST 12      PORT 'A' RELEASE W/ERRORS TEST
*
*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
*      BITS ARE SET IN THE DRIVE.
  
```



- \* A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
- \* B. WRITE 1'S INTO RMER1 THROUGH PORT 'A'.
- \* C. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RMER1 HAS NOT BEEN CLEARED.
- \* D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'A'.
- \* E. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

\*\*\*\*\*

016000  
 016000 005737 001300  
 016004 001406  
 016006 100002  
 016010 000137 003110  
 016014 012737 177777 001300  
 016022 012737 016036 001106  
 016030 012737 016036 001110  
 016036  
 016036 112737 000012 001102  
 016044 012706 001100  
 016050 012737 000031 001176

492  
 526

```
TST12:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOV      #TEST12,$LPADR ;SETUP SCOPE LOOP ADDRESS
        MOV      #TEST12,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST12:
        MOVB     #12,$STSTNM ;MOVE #12 TO TEST NUMBER
        MOV      #STACK,SP ;LOAD THE STACK POINTER
        MOV      #25, $TIMES ;DO 25. ITERATIONS
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

016056 113760 001224 000010  
 016064 005060 000012  
 016070 012760 000011 000000  
 016076 012760 000013 000000  
 016104 113760 001226 000010  
 016112 005060 000012  
 016116 012760 000011 000000  
 016124 012760 000013 000000

```
MOV      PORTA, RMCS2(R0) ;SELECT PORT #A
CLR      RMDS(R0)        ;SEIZE THE DRIVE
MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE
MOV      PORTB, RMCS2(R0) ;SELECT PORT #B
CLR      RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE
```

;SEIZE THE DRIVE THROUGH PORT A

016132 113760 001224 000010  
 016140 013737 001224 001242  
 016146 005060 000012  
 016152 013737 001226 001244

```
MOV      PORTA, RMCS2(R0) ;SELECT PORT A
MOV      PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR      RMDS(R0)        ;WRITE RMDS
MOV      PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
```

;FORCE AN ERROR

016160 012760 177777 000014  
 016166 012760 000013 000000  
 016174 005037 001250  
 016200 016037 000000 001126  
 016206 012737 000000 001122  
 016214 060037 001122  
 016220 012737 004012 001124  
 016226 013737 001126 001164

```
MOV      #-1, RMER1(R0) ;SET ERROR BITS
MOV      #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND
CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV      RMCS1(R0), $BDDAT ;GET CONTENTS OF RMCS1
MOV      #RMCS1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD      R0, $BDADR      ;ADD RH/RM BASE ADDRESS
MOV      #4012, $GDDAT ;WHAT REGISTER SHOULD BE
MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
```

```

016234 042737 173765 001164      BIC    #^C4012,$TMP0 ;SAVE SPECIFIED BITS
016242 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
016250 001414                      BEQ    66$           ;BR IF OK
016252 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
016260 042737 004012 001174      BIC    #4012,$TMP4  ;CLEAR THE MASKED BITS
016266 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
016274 104025                      EMT    25
016276 005137 001250                      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
016302 000240                      NOP
016304 005737 001250      66$:  TST    CKERR        ;DID 'GO' BIT RESET ?
016310 001002                      BNE    .+6         ;BR IF NOT
016312 000137 016352                      JMP    1$         ;'GO' BIT RESET
016316 012760 000040 000010      MOV    #CLR, RMCS2(RO) ;INIT THE RH/RM
016324 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
016332 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016340 012760 000013 000000      MOV    #13, RMCS1(RO) ;RELEASE THE DRIVE THROUGH PORT A
016346 000137 017116                      JMP    2$         ;BYPASS THE REST OF THE TEST
  
```

;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A

```

016352 113760 001226 000010      1$:  MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
016360 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016366 005037 001250                      CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
016372 016037 000012 001126      MOV    RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
016400 012737 000012 001122      MOV    #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016406 060037 001122                      ADD    RO, $BDADR    ;ADD RH/RM BASE ADDRESS
016412 005037 001124                      CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
016416 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
016424 001403                      BEQ    68$         ;BR IF OK
016426 104024                      EMT    24
016430 005137 001250                      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
016434 000240                      NOP
016436 113760 001224 000010      68$:  MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
016444 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016452 005037 001250                      CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
016456 016037 000014 001126      MOV    RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
016464 012737 000014 001122      MOV    #RMER1, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016472 060037 001122                      ADD    RO, $BDADR    ;ADD RH/RM BASE ADDRESS
016476 012737 177777 001124      MOV    #177777, $GDDAT ;WHAT REGISTER SHOULD BE
016504 023737 001124 001126      CMP    $GDDAT, $BDDAT ;IS THE REGISTER OK ?
016512 001403                      BEQ    70$         ;BR IF OK
016514 104010                      EMT    10
016516 005137 001250                      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
016522 000240                      NOP
  
```

;CLEAR THE ERRORS THROUGH PORT A

```

016524 012760 000011 000000      MOV    #11, RMCS1(RO) ;ISSUE A DRIVE CLEAR
  
```

;RELEASE THE DRIVE FROM PORT A

```

016532 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A
016540 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016546 012760 000013 000000      MOV    #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL



```

016554 005037 001254          CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
016560 012737 000012 001122  MOV      #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
016566 060037 001122          ADD      RO,$BDADR      ;ADD THE I/O BASE ADDRESS
016572 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
016600 113760 001224 000010  MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
016606 016037 000012 001170  MOV      RMDS(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
016614 042737 024001 001170  BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
016622 013737 001170 001164  MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
016630 042737 100100 001164  BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016636 113760 001226 000010  MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
016644 016037 000012 001172  MOV      RMDS(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
016652 042737 024001 001172  BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
016660 013737 001172 001166  MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
016666 042737 100100 001166  BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016674 023737 001164 001166  CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
016702 001006          BNE     72$             ;BR IF NOT
016704 005737 001164          TST     $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
016710 001045          BNE     74$             ;BR IF NOT
016712 104046          EMT     46
016714 000137 017114          JMP     76$             ;BYPASS THE REST OF THE CHECKS
016720 013737 001170 001126 72$:  MOV     $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
016726 013737 001226 001240  MOV     PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016734 113760 001226 000010  MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
016742 005737 001164          TST     $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
016746 001414          BEQ     73$             ;BR IF ZERO
016750 013737 001224 001240  MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016756 013737 001172 001126  MOV     $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
016764 113760 001224 000010  MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
016772 005737 001166          TST     $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
016776 001012          BNE     74$             ;BR IF NOT
017000 012737 177777 001254 73$:  MOV     #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
017006 012760 000011 000000  MOV     #11,RMCS1(RO)  ;CLEAR THE DRIVE
017014 012760 000013 000000  MOV     #13,RMCS1(RO)  ;RELEASE THE DRIVE
017022 104026          EMT     26
017024 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
017032 013737 001224 001240  MOV     PORTA,PTNBR    ;CHANGE PORT NUMBER
017040 042737 100000 001126  BIC     #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
017046 023737 001124 001126  CMP     $GDDAT,$BDDAT  ;ALL BITS OK ?
017054 001401          BEQ     75$             ;BR IF OK FROM PORT A.
017056 104007          EMT     7
017060 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
017066 013737 001226 001240  MOV     PORTB,PTNBR    ;CHANGE PORT NUMBER
017074 042737 100000 001126  BIC     #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
017102 023737 001124 001126  CMP     $GDDAT,$BDDAT  ;SEE IF READ OK FROM PORT B.
017110 001401          BEQ     76$             ;BR IF OK
017112 104007          EMT     7
017114 000240          76$:  NOP
017116 000004          2$:  SCOPE                ;LOOP ?

```

527  
547  
548

```

*****
;*TEST 13      PORT 'B' RELEASE W/ERRORS TEST
;*
;*VERIFY THAT A RELEASE COMMAND PERFORMS NO ACTION IF ISSUED WHEN ERROR
;*      BITS ARE SET IN THE DRIVE.
;*

```

- \* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- \* B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
- \* C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE 'GO' BIT HAS RESET, THAT THE DRIVE HAS NOT RETURNED TO NEUTRAL, AND THAT RMER1 HAS NOT BEEN CLEARED.
- \* D. CLEAR RMER1 BY ISSUING A DRIVE CLEAR COMMAND THROUGH PORT 'B'.
- \* E. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

\*\*\*\*\*  
 TST13:

017120  
 017120 005737 001300  
 017124 001406  
 017126 100002  
 017130 000137 003110  
 017134 012737 177777 001300  
 017142 012737 017156 001106  
 017150 012737 017156 001110  
 017156  
 017156 112737 000013 001102  
 017164 012706 001100  
 017170 012737 000031 001176

```

TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #TEST13,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST13,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST13:
MOVB #13,$STNM ;MOVE #13 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #25,$TIMES ;;DO 25. ITERATIONS
  
```

549  
 550

;CLEAR ATTENTION BITS FOR BOTH PORTS

017176 113760 001224 000010  
 017204 005060 000012  
 017210 012760 000011 000000  
 017216 012760 000013 000000  
 017224 113760 001226 000010  
 017232 005060 000012  
 017236 012760 000011 000000  
 017244 012760 000013 000000

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
CLR RMDS(R0) ;SEIZE THE DRIVE
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT B

017252 113760 001226 000010  
 017260 013737 001226 001242  
 017266 005060 000012  
 017272 013737 001224 001244

```

MOVB PORTB,RMCS2(R0) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
CLR RMDS(R0) ;WRITE RMDS
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
  
```

;FORCE AN ERROR

017300 012760 177777 000014  
 017306 012760 000013 000000  
 017314 005037 001250  
 017320 016037 000000 001126  
 017326 012737 000000 001122  
 017334 060037 001122  
 017340 012737 004012 001124  
 017346 013737 001126 001164  
 017354 042737 173765 001164

```

MOV #-1,RMER1(R0) ;SET ERROR BITS
MOV #13,RMCS1(R0) ;ISSUE A RELEASE COMMAND
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMCS1(R0),$BDDAT ;GET CONTENTS OF RMCS1
MOV #RMCS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
MOV #4012,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC #^C4012,$TMP0 ;SAVE SPECIFIED BITS
  
```



```

017362 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
017370 001414                      BEQ      66$              ;BR IF OK
017372 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
017400 042737 004012 001174      BIC      #4012,$TMP4      ;CLEAR THE MASKED BITS
017406 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
017414 104025                      EMT      25
017416 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERRUR INDICATOR
017422 000240                      66$:    NOP
017424 005737 001250                      TST      CKERR            ;DID 'GO' BIT RESET ?
017430 001002                      BNE      .+6              ;BR IF NOT
017432 000137 017472                      JMP      1$              ;'GO' BIT RESET
017436 012760 000040 000010      MOV      #CLR,RMCS2(RO)   ;INIT THE RH/RM
017444 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
017452 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017460 012760 000013 000000      MOV      #13,RMCS1(RO)   ;RELEASE THE DRIVE THROUGH PORT B
017466 000137 020236                      JMP      2$              ;BYPASS THE REST OF THE TEST
  
```

;VERIFY THAT DRIVE IS STILL SEIZED BY PORT B

```

017472 113760 001224 000010      1$:    MOV      PORTA,RMCS2(RO) ;SELECT PORT A
017472 013737 001224 001240      MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017500 005037 001250                      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
017506 016037 000012 001126      MOV      RMDS(RO),$BDDAT  ;GET CONTENTS OF RMDS
017512 012737 000012 001122      MOV      #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017520 060037 001122                      ADD      RO,$BDADR        ;ADD RH/RM BASE ADDRESS
017526 005037 001124                      CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
017532 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
017536 001403                      BEQ      68$              ;BR IF OK
017544 104024                      EMT      24
017550 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
017554 000240                      68$:    NOP
017556 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
017564 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017572 005037 001250                      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
017576 016037 000014 001126      MOV      RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
017604 012737 000014 001122      MOV      #RMER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
017612 060037 001122                      ADD      RO,$BDADR        ;ADD RH/RM BASE ADDRESS
017616 012737 177777 001124      MOV      #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
017624 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
017632 001403                      BEQ      70$              ;BR IF OK
017634 104010                      EMT      10
017636 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
017642 000240                      70$:    NOP
  
```

;CLEAR THE ERRORS THROUGH PORT B

```

017644 012760 000011 000000      MOV      #11,RMCS1(RO)   ;ISSUE A DRIVE CLEAR
  
```

;RELEASE THE DRIVE FROM PORT B

```

017652 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
017660 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
017666 012760 000013 000000      MOV      #13,RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

017674 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
017700 012737 000012 001122 MOV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
017706 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
017712 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
017720 113760 001224 000010 MOV#B PORTA, RMCS2(RO) ;SELECT PORT A.
017726 016037 000012 001170 MOV RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
017734 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
017742 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
017750 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
017756 113760 001226 000010 MOV#B PORTB, RMCS2(RO) ;SELECT PORT B.
017764 016037 000012 001172 MOV RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
017772 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
020000 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
020006 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
020014 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
020022 001006 BNE 72$ ;BR IF NOT
020024 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
020030 001045 BNE 74$ ;BR IF NOT
020032 104046 EMT 46
020034 000137 020234 JMP 76$ ;BYPASS THE REST OF THE CHECKS
020040 013737 001170 001126 72$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
020046 013737 001226 001240 MCV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020054 113760 001226 000010 MOV#B PORTB, RMCS2(RO) ;SELECT PORT B.
020062 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
020066 001414 BEQ 73$ ;BR IF ZERO
020070 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020076 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
020104 113760 001224 000010 MOV#B PORTA, RMCS2(RO) ;SELECT PORT A.
020112 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
020116 001012 BNE 74$ ;BR IF NOT
020120 012737 177777 001254 73$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
020126 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
020134 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
020142 104026 EMT 26
020144 013737 001170 001126 74$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
020152 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
020160 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
020166 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
020174 001401 BEQ 75$ ;BR IF OK FROM PORT A.
020176 104007 EMT 7
020200 013737 001172 001126 75$: MOV $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
020206 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
020214 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
020222 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
020230 001401 BEQ 76$ ;BR IF OK
020232 104007 EMT 7
020234 000240 76$: NOP
020236 000004 2$: SCOPE ;LOOP ?

```

551  
 570  
 571

```

*****
*TEST 14 PORT 'A' SEIZE AND CLEAR TEST
*
*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
* PORT TO RELEASE THE DRIVE.
*
* A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'A'.

```



- \* VERIFY THAT THE DRIVE HAS BEEN SEIZED.
- \* B. ISSUE A DRIVE CLEAR THROUGH PORT 'A' AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- \* C. ISSUE A MASSBUS CLEAR THROUGH THE RH/RM AND VERIFY THAT THE DRIVE DOES NOT RETURN TO NEUTRAL.
- \* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

\*\*\*\*\*

020240  
 020240 005737 001300  
 020244 001406  
 020246 100002  
 020250 000137 003110  
 020254 012737 177777 001300  
 020262 012737 020276 001106  
 020270 012737 020276 001110  
 020276  
 020276 112737 000014 001102  
 020304 012706 001100  
 020310 012737 000031 001176

TST14:  
 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?  
 BEQ 2\$ ;BR IF NOT  
 BPL 1\$ ;BR IF JUST ENTERED TEST  
 JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
 1\$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR  
 2\$: MOV #TEST14,\$LPADR ;SETUP SCOPE LOOP ADDRESS  
 MOV #TEST14,\$LPERR ;SETUP ERROR LOOP ADDRESS  
 TEST14:  
 MOVB #14,\$TSTNM ;MOVE #14 TO TEST NUMBER  
 MOV #STACK,SP ;LOAD THE STACK POINTER  
 MOV #25.,\$TIMES ;DO 25. ITERATIONS

572  
 602

;SEIZE THE DRIVE THROUGH PORT A

020316 113760 001224 000010  
 020324 013737 001224 001242  
 020332 005060 000012  
 020336 113760 001226 000010  
 020344 013737 001226 001240  
 020352 013737 001226 001244  
 020360 016037 000012 001126  
 020366 010037 001122  
 020372 062737 000012 001122  
 020400 005037 001124  
 020404 023737 001124 001126  
 020412 001403  
 020414 104004  
 020416 000137 021636  
 020422  
 020422 113760 001224 000010  
 020430 013737 001224 001240  
 020436 016037 000012 001126  
 020444 042737 020001 001126  
 020452 012737 011700 001124  
 020460 013737 001124 001166  
 020466 005137 001166  
 020472 013737 001126 001164  
 020500 043737 001166 001164  
 020506 023737 001124 001164  
 020514 001401  
 020516 104005  
 020520 000240

MOV B PORTA, RMCS2(R0) ;SELECT PORT A  
 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS  
 CLR RMDS(R0) ;WRITE RMDS  
 MOV B PORTB, RMCS2(R0) ;SELECT PORT B  
 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS  
 MOV RMDS(R0), \$BDDAT ;SEE IF DRIVE SEIZED BY PORT A  
 MOV R0, \$BDADR ;RH/RM BASE ADDRESS  
 ADD #RMDS, \$BDADR ;GENERATE BAD REGISTER ADDRESS  
 CLR \$GDDAT ;REGISTER SHOULD BE ZERO  
 CMP \$GDDAT, \$BDDAT ;IS THE REGISTER ZERO  
 BEQ 64\$ ;BR IF IT IS  
 EMT 4  
 JMP 1\$ ;BYPASS REST OF THE SUBTEST  
 64\$: MOV B PORTA, RMCS2(R0) ;SELECT PORT A  
 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 MOV RMDS(R0), \$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS  
 BIC #OM!PIP, \$BDDAT ;CLEAR DONT CARE BITS  
 MOV #MOL!PGM!DPR!DRY!VV, \$GDDAT ;EXPECTED STATUS  
 MOV \$GDDAT, \$TMP1 ;USE GOOD DATA AS A MASK  
 COM \$TMP1 ;COMPLEMENT THE EXPECTED STATUS  
 MOV \$BDDAT, \$TMP0 ;SAVE THE ACTUAL STATUS  
 BIC \$TMP1, \$TMP0 ;CLEAR UNWANTED BITS  
 CMP \$GDDAT, \$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?  
 BEQ 65\$ ;BR IF THEY ARE  
 EMT 5  
 65\$: NOP

```

;DRIVE CLEAR THROUGH PORT A FIRST
020522 012760 000011 000000      MOV      #11,RMCS1(RO) ;ISSUE DRIVE CLEAR THROUGH PORT A
;VERIFY THAT DRIVE STILL SEIZED BY PORT A
020530 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
020536 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020544 005037 001250                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
020550 016037 000012 001126      MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
020556 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020564 060037 001122                ADD      RO,$BDADR ;ADD RH/RM BASE ADDRESS
020570 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
020574 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
020602 042737 100000 001164      BIC      #^C77777,$TMP0 ;SAVE SPECIFIED BITS
020610 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
020616 001414                BEQ      66$ ;BR IF OK
020620 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
020626 042737 077777 001174      BIC      #77777,$TMP4 ;CLEAR THE MASKED BITS
020634 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
020642 104033                EMT      33
020644 005137 001250                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
020650 000240                NOP
020652 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
020660 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020666 005037 001250                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
020672 016037 000012 001126      MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
020700 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020706 060037 001122                ADD      RO,$BDADR ;ADD RH/RM BASE ADDRESS
020712 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
020720 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
020726 042737 100000 001164      BIC      #^C77777,$TMP0 ;SAVE SPECIFIED BITS
020734 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
020742 001414                BEQ      68$ ;BR IF OK
020744 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
020752 042737 077777 001174      BIC      #77777,$TMP4 ;CLEAR THE MASKED BITS
020760 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
020766 104033                EMT      33
020770 005137 001250                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
020774 000240                NOP
66$:
68$:

```

;NOW ISSUE MASSBUS INIT

```

020776 012760 000040 000010      MOV      #CLR,RMCS2(RO) ;ISSUE MASSBUS INIT

```

;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

```

021004 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B
021012 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021020 005037 001250                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
021024 016037 000012 001126      MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
021032 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021040 060037 001122                ADD      RO,$BDADR ;ADD RH/RM BASE ADDRESS
021044 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
021050 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
021056 042737 100000 001164      BIC      #^C77777,$TMP0 ;SAVE SPECIFIED BITS
021064 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS

```



```

021072 001414          BEQ      70$          ;BR IF OK
021074 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
021102 042737 077777 001174  BIC     #77777,$TMP4 ;CLEAR THE MASKED BITS
021110 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
021116 104034          EMT     34
021120 005137 001250          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
021124 000240          NOP
021126 113760 001224 000010 70$:  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
021134 013737 001224 001240  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021142 005037 001250          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
021146 016037 000012 001126  MOV     RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
021154 012737 000012 001122  MOV     #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021162 060037 001122          ADD     RO,$BDADR      ;ADD RH/RM BASE ADDRESS
021166 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
021174 013737 001126 001164  MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
021202 042737 100000 001164  BIC     #^C77777,$TMP0 ;SAVE SPECIFIED BITS
021210 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
021216 001414          BEQ      72$          ;BR IF OK
021220 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
021226 042737 077777 001174  BIC     #77777,$TMP4 ;CLEAR THE MASKED BITS
021234 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
021242 104034          EMT     34
021244 005137 001250          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
021250 000240          NOP
  
```

;RELEASE THE DRIVE FROM PORT A

```

021252 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
021260 013737 001224 001240  MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021266 012760 000013 000000  MOV     #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

021274 005037 001254          CLR     RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
021300 012737 000012 001122  MOV     #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
021306 060037 001122          ADD     RO,$BDADR      ;ADD THE I/O BASE ADDRESS
021312 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
021320 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
021326 016037 000012 001170  MOV     RMDS(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
021334 042737 024001 001170  BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
021342 013737 001170 001164  MOV     $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
021350 042737 100100 001164  BIC     #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021356 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
021364 016037 000012 001172  MOV     RMDS(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
021372 042737 024001 001172  BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
021400 013737 001172 001166  MOV     $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
021406 042737 100100 001166  BIC     #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021414 023737 001164 001166  CMP     $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
021422 001006          BNE     74$          ;BR IF NOT
021424 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
021430 001045          BNE     76$          ;BR IF NOT
021432 104046          EMT     46
021434 000137 021634          JMP     78$          ;BYPASS THE REST OF THE CHECKS
021440 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
021446 013737 001226 001240  MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021454 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
021462 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
  
```

```

021466 001414 BEQ 75$ ;BR IF ZERO
021470 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021476 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
021504 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
021512 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
021516 001012 BNE 76$ ;BR IF NOT
021520 012737 177777 001254 75$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
021526 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
021534 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
021542 104026 EMT 26
021544 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
021552 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
021560 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
021566 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
021574 001401 BEQ 77$ ;BR IF OK FROM PORT A.
021576 104007 EMT 7
021600 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
021606 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
021614 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
021622 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
021630 001401 BEQ 78$ ;BR IF OK
021632 104007 EMT 7
021634 000240 78$: NOP
021636 000004 1$: SCOPE ;LOOP ?
  
```

603  
622  
623

```

:*****
:*TEST 15 PORT 'B' SEIZE AND CLEAR TEST
:*
:*VERIFY THAT A MASSBUS CLEAR OR DRIVE CLEAR WILL NOT CAUSE THE SEIZING
:* PORT TO RELEASE THE DRIVE.
:*
:* A. SEIZE THE DRIVE BY WRITING 0'S INTO RMDS THROUGH PORT 'B'.
:* VERIFY THAT THE DRIVE HAS BEEN SEIZED.
:*
:* B. ISSUE A DRIVE CLEAR THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
:* DOES NOT RETURN TO NEUTRAL.
:*
:* C. ISSUE A MASSBUS CLEAR THROUGH THE RH/RM AND VERIFY THAT THE DRIVE
:* DOES NOT RETURN TO NEUTRAL.
:*
:* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE
:* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*
:*****
  
```

```

021640 TST15:
021640 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
021644 001406 BEQ 2$ ;BR IF NOT
021646 100002 BPL 1$ ;BR IF JUST ENTERED TEST
021650 000137 003110 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
021654 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
021662 012737 021676 001106 2$: MOV #TEST15,$LPADR ;SETUP SCOPE LOOP ADDRESS
021670 012737 021676 001110 MOV #TEST15,$LPERR ;SETUP ERROR LOOP ADDRESS
021676 TEST15:
021676 112737 000015 001102 MOVB #15,$STSTNM ;MOVE #15 TO TEST NUMBER
021704 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
021710 012737 000031 001176 MOV #25.,$TIMES ;DO 25. ITERATIONS
  
```



624  
625

;SEIZE THE DRIVE THROUGH PORT B

021716	113760	001226	000010	MOV	PORTB, RMCS2(R0)	;SELECT PORT B
021724	013737	001226	001242	MOV	PORTB, SEIZPT	;STORE SEIZING PORT'S ADDRESS
021732	005060	000012		CLR	RMDS(R0)	;WRITE RMDS
021736	113760	001224	000010	MOV	PORTA, RMCS2(R0)	;SELECT PORT A
021744	013737	001224	001240	MOV	PORTA, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021752	013737	001224	001244	MOV	PORTA, OPPRT	; 'OPPOSITE' PORT ADDRESS
021760	016037	000012	001126	MOV	RMDS(R0), \$BDDAT	;SEE IF DRIVE SEIZED BY PORT B
021766	010037	001122		MOV	R0, \$BDADR	;RH/RM BASE ADDRESS
021772	062737	000012	001122	ADD	#RMDS, \$BDADR	;GENERATE BAD REGISTER ADDRESS
022000	005037	001124		CLR	\$GDDAT	;REGISTER SHOULD BE ZERO
022004	023737	001124	001126	CMP	\$GDDAT, \$BDDAT	;IS THE REGISTER ZERO
022012	001403			BEQ	64\$	;BR IF IT IS
022014	104004			EMT	4	
022016	000137	023236		JMP	1\$	;BYPASS REST OF THE SUBTEST
022022						
022022	113760	001226	000010	MOV	PORTB, RMCS2(R0)	;SELECT PORT B
022030	013737	001226	001240	MOV	PORTB, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022036	016037	000012	001126	MOV	RMDS(R0), \$BDDAT	;SEE IF SEIZING PORT SEES CORRECT STATUS
022044	042737	020001	001126	BIC	#OM!PIP, \$BDDAT	;CLEAR DONT CARE BITS
022052	012737	011700	001124	MOV	#MOL!PGM!DPR!DRY!VV, \$GDDAT	;EXPECTED STATUS
022060	013737	001124	001166	MOV	\$GDDAT, \$TMP1	;USE GOOD DATA AS A MASK
022066	005137	001166		COM	\$TMP1	;COMPLEMENT THE EXPECTED STATUS
022072	013737	001126	001164	MOV	\$BDDAT, \$TMP0	;SAVE THE ACTUAL STATUS
022100	043737	001166	001164	BIC	\$TMP1, \$TMP0	;CLEAR UNWANTED BITS
022106	023737	001124	001164	CMP	\$GDDAT, \$TMP0	;ARE THE EXPECTED STATUS BITS SET ?
022114	001401			BEQ	65\$	;BR IF THEY ARE
022116	104005			EMT	5	
022120	000240			NOP		

;DRIVE CLEAR THROUGH PORT B FIRST

022122	012760	000011	000000	MOV	#11, RMCS1(R0)	;ISSUE DRIVE CLEAR THROUGH PORT B
--------	--------	--------	--------	-----	----------------	-----------------------------------

;VERIFY THAT DRIVE STILL SEIZED BY PORT B

022130	113760	001224	000010	MOV	PORTA, RMCS2(R0)	;SELECT PORT A
022136	013737	001224	001240	MOV	PORTA, PTNBR	;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022144	005037	001250		CLR	CKERR	;CLEAR THE 'CHECK ERROR' INDICATOR
022150	016037	000012	001126	MOV	RMDS(R0), \$BDDAT	;GET CONTENTS OF RMDS
022156	012737	000012	001122	MOV	#RMDS, \$BDADR	;FORM REGISTER ADDRESS OF ERROR MESSAGE
022164	060037	001122		ADD	R0, \$BDADR	;ADD RH/RM BASE ADDRESS
022170	005037	001124		CLR	\$GDDAT	;WHAT REGISTER SHOULD BE
022174	013737	001126	001164	MOV	\$BDDAT, \$TMP0	;MOVE REGISTER CONTENTS TO '\$TMP0'
022202	042737	100000	001164	BIC	#^C77777, \$TMP0	;SAVE SPECIFIED BITS
022210	023737	001124	001164	CMP	\$GDDAT, \$TMP0	;COMPARE THE BITS
022216	001414			BEQ	66\$	;BR IF OK
022220	013737	001126	001174	MOV	\$BDDAT, \$TMP4	;COPY 'BAD DATA'
022226	042737	077777	001174	BIC	#77777, \$TMP4	;CLEAR THE MASKED BITS
022234	053737	001174	001124	BIS	\$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
022242	104033			EMT	33	
022244	005137	001250		COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
022250	000240			NOP		
022252	113760	001226	000010	MOV	PORTB, RMCS2(R0)	;SELECT PORT B



```

022260 013737 001226 001240      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022266 005037 001250                CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
022272 016037 000012 001126      MOV    RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
022300 012737 000012 001122      MOV    #RMD5,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022306 060037 001122                ADD    R0,$BDADR    ;ADD RH/RM BASE ADDRESS
022312 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
022320 013737 001126 001164      MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
022326 042737 100000 001164      BIC    #^C77777,$TMP0 ;SAVE SPECIFIED BITS
022334 023737 001124 001164      CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
022342 001414                BEQ    68$          ;BR IF OK
022344 013737 001126 001174      MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
022352 042737 077777 001174      BIC    #77777,$TMP4  ;CLEAR THE MASKED BITS
022360 053737 001174 001124      BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
022366 104033                EMT    33
022370 005137 001250                COM    CKERR
022374 000240                NOP
  
```

68\$: ;SET THE REGISTER COMPARE ERROR INDICATOR

;NOW ISSUE MASSBUS INIT

```

022376 012760 000040 000010      MOV    #CLR,RMCS2(R0) ;ISSUE MASSBUS INIT
  
```

;CONFIRM THAT DRIVE STILL SEIZED BY PORT B

```

022404 113760 001224 000010      MOV    PORTA,RMCS2(R0) ;SELECT PORT A
022412 013737 001224 001240      MOV    PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022420 005037 001250                CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
022424 016037 000012 001126      MOV    RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
022432 012737 000012 001122      MOV    #RMD5,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022440 060037 001122                ADD    R0,$BDADR    ;ADD RH/RM BASE ADDRESS
022444 005037 001124                CLR    $GDDAT      ;WHAT REGISTER SHOULD BE
022450 013737 001126 001164      MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
022456 042737 100000 001164      BIC    #^C77777,$TMP0 ;SAVE SPECIFIED BITS
022464 023737 001124 001164      CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
022472 001414                BEQ    70$          ;BR IF OK
022474 013737 001126 001174      MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
022502 042737 077777 001174      BIC    #77777,$TMP4  ;CLEAR THE MASKED BITS
022510 053737 001174 001124      BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
022516 104034                EMT    34
022520 005137 001250                COM    CKERR
022524 000240                NOP
  
```

70\$: ;SET THE REGISTER COMPARE ERROR INDICATOR

```

022526 113760 001226 000010      MOV    PORTB,RMCS2(R0) ;SELECT PORT B
022534 013737 001226 001240      MOV    PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022542 005037 001250                CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
022546 016037 000012 001126      MOV    RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
022554 012737 000012 001122      MOV    #RMD5,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022562 060037 001122                ADD    R0,$BDADR    ;ADD RH/RM BASE ADDRESS
022566 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
022574 013737 001126 001164      MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
022602 042737 100000 001164      BIC    #^C77777,$TMP0 ;SAVE SPECIFIED BITS
022610 023737 001124 001164      CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
022616 001414                BEQ    72$          ;BR IF OK
022620 013737 001126 001174      MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
022626 042737 077777 001174      BIC    #77777,$TMP4  ;CLEAR THE MASKED BITS
022634 053737 001174 001124      BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
022642 104034                EMT    34
022644 005137 001250                COM    CKERR
022650 000240                NOP
  
```

72\$: ;SET THE REGISTER COMPARE ERROR INDICATOR



;RELEASE THE DRIVE FROM PORT B

```
022652 113760 001226 000010      MOV# PORTB, RMCS2(R0) ;SELECT PORT B
022660 013737 001226 001240      MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022666 012760 000013 000000      MOV  #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
022674 005037 001254      CLR  RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
022700 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
022706 060037 001122      ADD  R0, $BDADR ;ADD THE I/O BASE ADDRESS
022712 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
022720 113760 001224 000010      MOV# PORTA, RMCS2(R0) ;SELECT PORT A.
022726 016037 000012 001170      MOV  RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022734 042737 024001 001170      BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
022742 013737 001170 001164      MOV  $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
022750 042737 100100 001164      BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022756 113760 001226 000010      MOV# PORTB, RMCS2(R0) ;SELECT PORT B.
022764 016037 000012 001172      MOV  RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022772 042737 024001 001172      BIC  #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
023000 013737 001172 001166      MOV  $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
023006 042737 100100 001166      BIC  #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023014 023737 001164 001166      CMP  $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023022 001006      BNE  74$ ;BR IF NOT
023024 005737 001164      TST  $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
023030 001045      BNE  76$ ;BR IF NOT
023032 104046      EMT  46
023034 000137 023234      JMP  78$ ;BYPASS THE REST OF THE CHECKS
023040 013737 001170 001126 74$: MOV  $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023046 013737 001226 001240      MOV  PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023054 113760 001226 000010      MOV# PORTB, RMCS2(R0) ;SELECT PORT B.
023062 005737 001164      TST  $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
023066 001414      BEQ  75$ ;BR IF ZERO
023070 013737 001224 001240      MOV  PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023076 013737 001172 001126      MOV  $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
023104 113760 001224 000010      MOV# PORTA, RMCS2(R0) ;SELECT PORT A.
023112 005737 001166      TST  $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
023116 001012      BNE  76$ ;BR IF NOT
023120 012737 177777 001254 75$: MOV  #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
023126 012760 000011 000000      MOV  #11, RMCS1(R0) ;CLEAR THE DRIVE
023134 012760 000013 000000      MOV  #13, RMCS1(R0) ;RELEASE THE DRIVE
023142 104026      EMT  26
023144 013737 001170 001126 76$: MOV  $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
023152 013737 001224 001240      MOV  PORTA, PTNBR ;CHANGE PORT NUMBER
023160 042737 100000 001126      BIC  #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
023166 023737 001124 001126      CMP  $GDDAT, $BDDAT ;ALL BITS OK ?
023174 001401      BEQ  77$ ;BR IF OK FROM PORT A.
023176 104007      EMT  7
023200 013737 001172 001126 77$: MOV  $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
023206 013737 001226 001240      MOV  PORTB, PTNBR ;CHANGE PORT NUMBER
023214 042737 100000 001126      BIC  #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
023222 023737 001124 001126      CMP  $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
023230 001401      BEQ  78$ ;BR IF OK
023232 104007      EMT  7
023234 000240      NOP
023236 000004      1$: SCOPE ;LOOP ?
```



626  
 638  
 639

```

*****
*TEST 16 SEIZE 'A' BY RMCS1 TEST
*
*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
*IF THE DRIVE IS IN NEUTRAL.
* A. READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'A'; VERIFY THAT
* THE DRIVE IS SEIZED.
*
* B. ISSUE A RELEASE COMMAND THROUGH PORT 'A'; VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****
  
```

```

023240
023240 005737 001300
023244 001406
023246 100002
023250 000137 003110
023254 012737 177777 001300
023262 012737 023276 001106
023270 012737 023276 001110
023276
023276 112737 000016 001102
023304 012706 001100
023310 012737 000031 001176
  
```

```

TST16:
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #TEST16,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST16,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST16:
MOVB #16,$STSTM ;MOVE #16 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #25,$TIMES ;;DO 25. ITERATIONS
  
```

640  
 650

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

023316 113760 001224 000010
023324 005060 000012
023330 012760 000011 000000
023336 012760 000013 000000
023344 113760 001226 000010
023352 005060 000012
023356 012760 000011 000000
023364 012760 000013 000000
  
```

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
CLR RMDS(R0) ;SEIZE THE DRIVE
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

023372 113760 001224 000010
023400 013737 001224 001242
023406 005760 000000
023412 113760 001226 000010
023420 013737 001226 001240
023426 013737 001226 001244
023434 016037 000012 001126
023442 010037 001122
023446 062737 000012 001122
023454 005037 001124
023460 023737 001124 001126
023466 001403
023470 104004
023472 000137 024162
023476
023476 113760 001224 000010
  
```

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RMCS1(R0) ;READ RMCS1
MOVB PORTB,RMCS2(R0) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
MOV RO,$BDADR ;RH/RM BASE ADDRESS
ADD #RMDS,$BDADR ;GENERATE BAD REGISTER ADDRESS
CLR $GDDAT ;REGISTER SHOULD BE ZERO
CMP $GDDAT,$BDDAT ;IS THE REGISTER ZERO
BEQ 64$ ;BR IF IT IS
EMT 4
JMP 1$ ;BYPASS REST OF THE SUBTEST

64$: MOVB PORTA,RMCS2(R0) ;SELECT PORT A
  
```



```

023504 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
023512 016037 000012 001126      MOV      RMDS(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
023520 042737 020001 001126      BIC      #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
023526 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
023534 013737 001124 001166      MOV      $GDDAT,$TMP1 ;USE GOOD DATA AS A MASK
023542 005137 001166                COM      $TMP1 ;COMPLEMENT THE EXPECTED STATUS
023546 013737 001126 001164      MOV      $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
023554 043737 001166 001164      BIC      $TMP1,$TMP0 ;CLEAR UNWANTED BITS
023562 023737 001124 001164      CMP      $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
023570 001401                BEQ      65$ ;BR IF THEY ARE
023572 104005                EMT      5
023574 000240                NOP
  
```

65\$:

;RELEASE THE DRIVE FROM PORT A

```

023576 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A
023604 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
023612 012760 000013 000000      MOV      #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

023620 005037 001254                CLR      RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
023624 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
023632 060037 001122                ADD      RO,$BDADR ;ADD THE I/O BASE ADDRESS
023636 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
023644 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
023652 016037 000012 001170      MOV      RMDS(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
023660 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
023666 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
023674 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023702 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
023710 016037 000012 001172      MOV      RMDS(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
023716 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
023724 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
023732 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023740 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023746 001006                BNE      66$ ;BR IF NOT
023750 005737 001164                TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
023754 001045                BNE      68$ ;BR IF NOT
023756 104046                EMT      46
023760 000137 024160                JMP      70$ ;BYPASS THE REST OF THE CHECKS
023764 013737 001170 001126 66$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023772 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024000 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
024006 005737 001164                TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
024012 001414                BEQ      67$ ;BR IF ZERO
024014 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024022 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
024030 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
024036 005737 001166                TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
024042 001012                BNE      68$ ;BR IF NOT
024044 012737 177777 001254 67$: MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
024052 012760 000011 000000      MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
024060 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
024066 104026                EMT      26
024070 013737 001170 001126 68$: MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
024076 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
  
```

```

024104 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
024112 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;ALL BITS OK ?
024120 001401                      BEQ      69$             ;BR IF OK FROM PORT A.
024122 104007                      EMT      7
024124 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
024132 013737 001226 001240      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
024140 042737 100000 001126      BIC      #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
024146 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;SEE IF READ OK FROM PORT B.
024154 001401                      BEQ      70$             ;BR IF OK
024156 104007                      EMT      7
024160 000240                      70$:    NOP
024162 000004                      1$:     SCOPE           ;LOOP ?
  
```

651  
663  
664

```

:*****
:*TEST 17      SEIZE 'B' BY RMCS1 TEST
:*
:*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) SEIZES THE DRIVE
:*IF THE DRIVE IS IN NEUTRAL.
:*  A.  READ THE CONTROL REGISTER (RMCS1) THROUGH PORT 'B'; VERIFY THAT
:*      THE DRIVE IS SEIZED.
:*
:*  B.  ISSUE A RELEASE COMMAND THROUGH PORT 'B'; VERIFY THAT THE DRIVE
:*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*****
  
```

```

024164 005737 001300      TST      KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
024170 001406                      BEQ      2$              ;BR IF NOT
024172 100002                      BPL      1$              ;BR IF JUST ENTERED TEST
024174 000137 003110      JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
024200 012737 177777 001300 1$:    MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
024206 012737 024222 001106 2$:    MOV      #TEST17,$LPADR ;SETUP SCOPE LOOP ADDRESS
024214 012737 024222 001110      MOV      #TEST17,$LPERR ;SETUP ERROR LOOP ADDRESS
024222                      TEST17:
024222 112737 000017 001102      MOV      #17,$STSTNM    ;MOVE #17 TO TEST NUMBER
024230 012706 001100      MOV      #STACK,SP      ;LOAD THE STACK POINTER
024234 012737 000031 001176      MOV      #25.,$TIMES    ;DO 25. ITERATIONS
  
```

665  
666

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

024242 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT #A
024250 005060 000012      CLR      RMDS(R0)        ;SEIZE THE DRIVE
024254 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
024262 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
024270 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT #B
024276 005060 000012      CLR      RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
024302 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
024310 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

024316 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
024324 013737 001226 001242      MOV      PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
024332 005760 000000      TST      RMCS1(R0)      ;READ RMCS1
024336 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
  
```



```

024344 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024352 013737 001224 001244      MOV     PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
024360 016037 000012 001126      MOV     RMDS(R0),%BDDAT ;SEE IF DRIVE SEIZED BY PORT B
024366 010037 001122                MOV     R0,%BDADR ;RH/RM BASE ADDRESS
024372 062737 000012 001122      ADD     #RMDS,%BDADR ;GENERATE BAD REGISTER ADDRESS
024400 005037 001124                CLR     %GDDAT ;REGISTER SHOULD BE ZERO
024404 023737 001124 001126      CMP     %GDDAT,%BDDAT ;IS THE REGISTER ZERO
024412 001403                BEQ     64$ ;BR IF IT IS
024414 104004                EMT     4
024416 000137 025106                JMP     1$ ;BYPASS REST OF THE SUBTEST
024422                64$:
024422 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B
024430 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024436 016037 000012 001126      MOV     RMDS(R0),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
024444 042737 020001 001126      BIC     #OM!PIP,%BDDAT ;CLEAR DONT CARE BITS
024452 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,%GDDAT ;EXPECTED STATUS
024460 013737 001124 001166      MOV     %GDDAT,%TMP1 ;USE GOOD DATA AS A MASK
024466 005137 001166                COM     %TMP1 ;COMPLEMENT THE EXPECTED STATUS
024472 013737 001126 001164      MOV     %BDDAT,%TMP0 ;SAVE THE ACTUAL STATUS
024500 043737 001166 001164      BIC     %TMP1,%TMP0 ;CLEAR UNWANTED BITS
024506 023737 001124 001164      CMP     %GDDAT,%TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
024514 001401                BEQ     65$ ;BR IF THEY ARE
024516 104005                EMT     5
024520 000240                65$:
                                NOP

                                ;RELEASE THE DRIVE FROM PORT B

024522 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B
024530 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024536 012760 000013 000000      MOV     #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

024544 005037 001254                CLR     RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
024550 012737 000012 001122      MOV     #RMDS,%BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
024556 060037 001122                ADD     R0,%BDADR ;ADD THE I/O BASE ADDRESS
024562 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,%GDDAT ;COMPARISON CONSTANT
024570 113760 001224 000010      MOV     PORTA,RMCS2(R0) ;SELECT PORT A.
024576 016037 000012 001170      MOV     RMDS(R0),%TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
024604 042737 024001 001170      BIC     #PIP!WRL!OM,%TMP2 ;CLEAR DONT CARES
024612 013737 001170 001164      MOV     %TMP2,%TMP0 ;COPY IT INTO '%TMP0'
024620 042737 100100 001164      BIC     #ATA!VV,%TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024626 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B.
024634 016037 000012 001172      MOV     RMDS(R0),%TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
024642 042737 024001 001172      BIC     #PIP!WRL!OM,%TMP3 ;CLEAR DONT CARES
024650 013737 001172 001166      MOV     %TMP3,%TMP1 ;COPY IT INTO '%TMP1'
024656 042737 100100 001166      BIC     #ATA!VV,%TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024664 023737 001164 001166      CMP     %TMP0,%TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
024672 001006                BNE     66$ ;BR IF NOT
024674 005737 001164                TST     %TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
024700 001045                BNE     68$ ;BR IF NOT
024702 104046                EMT     46
024704 000137 025104                JMP     70$ ;BYPASS THE REST OF THE CHECKS
024710 013737 001170 001126      MOV     %TMP2,%BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
024716 013737 001226 001240      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024724 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B.
024732 005737 001164                TST     %TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.

```

117 SEIZE 'B' BY RMCS1 TEST

```

024736 001414
024740 013737 001224 001240
024746 013737 001172 001126
024754 113760 001224 000010
024762 005737 001166
024766 001012
024770 012737 177777 001254
024776 012760 000011 000000
025004 012760 000013 000000
025012 104026
025014 013737 001170 001126
025022 013737 001224 001240
025030 042737 100000 001126
025036 023737 001124 001126
025044 001401
025046 104007
025050 013737 001172 001126
025056 013737 001226 001240
025064 042737 100000 001126
025072 023737 001124 001126
025100 001401
025102 104007
025104 000240
025106 000004
    
```

```

BEQ 67$ ;BR IF ZERO
MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
MOV PORTA,RMCS2(RO) ;SELECT PORT A.
TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
BNE 68$ ;BR IF NOT
MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
EMT 26
MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
MOV PORTA,PTNBR ;CHANGE PORT NUMBER
BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
CMP $GDDAT,$BDDAT ;ALL BITS OK ?
BEQ 69$ ;BR IF OK FROM PORT A.
EMT 7
MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
MOV PORTB,PTNBR ;CHANGE PORT NUMBER
BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
BEQ 70$ ;BR IF OK
EMT 7
NOP
1$: SCOPE ;LOOP ?
    
```

667  
683  
684

```

*****
*TEST 20 PORT 'A' INHIBIT SEIZE BY RMCS1 TEST
*****
*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
*REQUEST' IF THE DRIVE IS SEIZED.
*
* A. SEIZE THE DRIVE THROUGH PORT 'B' BY READING RMCS1. VERIFY THAT
* THE DRIVE HAS BEEN SEIZED.
*
* B. READ THE CONTROL REGISTER FROM PORT 'A'. VERIFY THAT 'DVA' IS NOT
* SET.
*
* C. ISSUE A RELEASE COMMAND THROUGH PORT 'B'. VERIFY THAT THE DRIVE
* RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
    
```

```

025110
025110 005737 001300
025114 001406
025116 100002
025120 000137 003110
025124 012737 177777 001300
025132 012737 025146 001106
025140 012737 025146 001110
025146
025146 112737 000020 001102
025154 012706 001100
025160 012737 000031 001176
    
```

```

TST20:
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #TEST20,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST20,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST20:
MOV #20,$TSTNM ;MOVE #20 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #25, $TIMES ;DO 25. ITERATIONS
    
```

685  
702

;CLEAR ATTENTION BITS FOR BOTH PORTS



```

025166 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT #A
025174 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE
025200 012760 000011 000000      MOV   #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
025206 012760 000013 000000      MOV   #13,RMCS1(R0)  ;RELEASE THE DRIVE
025214 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT #B
025222 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
025226 012760 000011 000000      MOV   #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
025234 012760 000013 000000      MOV   #13,RMCS1(R0)  ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

025242 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
025250 013737 001226 001242      MOV   PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
025256 005760 000000              TST   RMCS1(R0)      ;READ RMCS1
025262 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
025270 013737 001224 001240      MOV   PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025276 013737 001224 001244      MOV   PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
025304 016037 000012 001126      MCV   RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
025312 010037 001122              MOV   RO,$BDADR      ;RH/RM BASE ADDRESS
025316 062737 000012 001122      ADD   #RMDS,$BDADR   ;GENERATE BAD REGISTER ADDRESS
025324 005037 001124              CLR   $GDDAT         ;REGISTER SHOULD BE ZERO
025330 023737 001124 001126      CMP   $GDDAT,$BDDAT  ;IS THE REGISTER ZERO
025336 001403                      BEQ   64$            ;BR IF IT IS
025340 104004                      EMT   4
025342 000137 026154              JMP   1$             ;BYPASS REST OF THE SUBTEST
  
```

64\$:

```

025346 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
025354 013737 001226 001240      MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025362 016037 000012 001126      MOV   RMDS(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
025370 042737 020001 001126      BIC   #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
025376 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
025404 013737 001124 001166      MOV   $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
025412 005137 001166              COM   $TMP1          ;COMPLEMENT THE EXPECTED STATUS
025416 013737 001126 001164      MOV   $BDDAT,$TMP0   ;SAVE THE ACTUAL STATUS
025424 043737 001166 001164      BIC   $TMP1,$TMP0    ;CLEAR UNWANTED BITS
025432 023737 001124 001164      CMP   $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
025440 001401                      BEQ   65$            ;BR IF THEY ARE
025442 104005                      EMT   5
  
```

65\$:

```

025446 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
025454 013737 001224 001240      MOV   PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

;READ RMCS1 THROUGH PORT A - TRY TO SET PORT REQUEST

```

025462 005037 001250              CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
025466 016037 000000 001126      MOV   RMCS1(R0),$BDDAT ;GET CONTENTS OF RMCS1
025474 012737 000000 001122      MOV   #RMCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
025502 060037 001122              ADD   RO,$BDADR      ;ADD RH/RM BASE ADDRESS
025506 005037 001124              CLR   $GDDAT         ;WHAT REGISTER SHOULD BE
025512 013737 001126 001164      MOV   $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
025520 042737 173700 001164      BIC   #^C4077,$TMP0  ;SAVE SPECIFIED BITS
025526 023737 001124 001164      CMP   $GDDAT,$TMP0   ;COMPARE THE BITS
025534 001414                      BEQ   66$            ;BR IF OK
025536 013737 001126 001174      MOV   $BDDAT,$TMP4   ;COPY 'BAD DATA'
025544 042737 004077 001174      BIC   #4077,$TMP4    ;CLEAR THE MASKED BITS
025552 053737 001174 001124      BIS   $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
  
```

```

025560 104010          EMT      10
025562 005137 001250  COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
025566 000240          NOP
66$:
;DRIVE SHOULD RETURN TO NEUTRAL
;RELEASE THE DRIVE FROM PORT B

025570 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
025576 013737 001226 001240      MCV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025604 012760 000013 000000      MOV   #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

025612 005037 001254          CLR   RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
025616 012737 000012 001122      MOV   #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
025624 060037 001122          ADD   R0, $BDADR  ;ADD THE I/O BASE ADDRESS
025630 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
025636 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
025644 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
025652 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
025660 013737 001170 001164      MOV   $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
025666 042737 100100 001164      BIC   #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025674 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
025702 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
025710 042737 024001 001172      BIC   #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
025716 013737 001172 001166      MOV   $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
025724 042737 100100 001166      BIC   #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025732 023737 001164 001166      CMP   $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
025740 001006          BNE   68$        ;BR IF NOT
025742 005737 001164          TST   $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
025746 001045          BNE   70$        ;BR IF NOT
025750 104046          EMT      46
025752 000137 026152          JMP   72$        ;BYPASS THE REST OF THE CHECKS
025756 013737 001170 001126 68$:  MOV   $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
025764 013737 001226 001240      MOV   PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025772 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
026000 005737 001164          TST   $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
026004 001414          BEQ   69$        ;BR IF ZERO
026006 013737 001224 001240      MOV   PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
026014 013737 001172 001126      MOV   $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
026022 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
026030 005737 001166          TST   $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
026034 001012          BNE   70$        ;BR IF NOT
026036 012737 177777 001254 69$:  MOV   #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
026044 012760 000011 000000      MOV   #11, RMCS1(R0) ;CLEAR THE DRIVE
026052 012760 000013 000000      MOV   #13, RMCS1(R0) ;RELEASE THE DRIVE
026060 104026          EMT      26
026062 013737 001170 001126 70$:  MOV   $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
026070 013737 001224 001240      MOV   PORTA, PTNBR ;CHANGE PORT NUMBER
026076 042737 100000 001126      BIC   #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
026104 023737 001124 001126      CMP   $GDDAT, $BDDAT ;ALL BITS OK ?
026112 001401          BEQ   71$        ;BR IF OK FROM PORT A.
026114 104007          EMT      7
026116 013737 001172 001126 71$:  MOV   $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
026124 013737 001226 001240      MOV   PORTB, PTNBR ;CHANGE PORT NUMBER
026132 042737 100000 001126      BIC   #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT

```



703  
719  
720

```

026140 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
026146 001401                      BEQ      72$           ;BR IF OK
026150 104007                      EMT      7            ;
026152 000240                      72$:    NOP
026154 000004                      1$:    SCOPE           ;LOOP ?
    
```

```

:*****
:*TEST 21      PORT 'B' INHIBIT SEIZE BY RMCS1 TEST
:*
:*VERIFY THAT READING THE CONTROL REGISTER (RMCS1) DOES NOT SET 'PORT
:*REQUEST' IF THE DRIVE IS SEIZED.
:*
:*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY READING RMCS1.  VERIFY THAT
:*      THE DRIVE HAS BEEN SEIZED.
:*
:*  B.  READ THE CONTROL REGISTER FROM PORT 'B'.  VERIFY THAT 'DVA' IS NOT
:*      SET.
:*
:*  C.  ISSUE A RELEASE COMMAND THROUGH PORT 'A'.  VERIFY THAT THE DRIVE
:*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*
:*****
    
```

721  
722

```

026156 005737 001300
026162 001406
026164 100002
026166 000137 003110
026172 012737 177777 001300 1$:  MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
026200 012737 026214 001106 2$:  MOV      #TEST21,$LPADR ;SETUP SCOPE LOOP ADDRESS
026206 012737 026214 001110      MOV      #TEST21,$LPERR ;SETUP ERROR LOOP ADDRESS
026214
026214 112737 000021 001102 TEST21: MOVB    #21,$TSTNM ;MOVE #21 TO TEST NUMBER
026222 012706 001100      MOV      #STACK,SP ;LOAD THE STACK POINTER
026226 012737 000031 001176      MOV      #25,,$TIMES ;DO 25. ITERATIONS
    
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

026234 113760 001224 000010      MOVB    PORTA,RMCS2(R0) ;SELECT PORT #A
026242 005060 000012      CLR     RMD5(R0)        ;SEIZE THE DRIVE
026246 012760 000011 000000      MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
026254 012760 000013 000000      MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
026262 113760 001226 000010      MOVB    PORTB,RMCS2(R0) ;SELECT PORT #B
026270 005060 000012      CLR     RMD5(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
026274 012760 000011 000000      MOV     #11,RMCS1(R0)   ;ISSUE DRIVE CLEAR
026302 012760 000013 000000      MOV     #13,RMCS1(R0)   ;RELEASE THE DRIVE
    
```

;SEIZE THE DRIVE THROUGH PORT A

```

026310 113760 001224 000010      MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
026316 013737 001224 001242      MOV     PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
026324 005760 000000      TST     RMCS1(R0)       ;READ RMCS1
026330 113760 001226 000010      MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
026336 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026344 013737 001226 001244      MOV     PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
026352 016037 000012 001126      MOV     RMD5(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT A
    
```

```

026360 010037 001122          MOV    RO,$BDADR      ;RH/RM BASE ADDRESS
026364 062737 000012 001122  ADD    #RMDS,$BDADR  ;GENERATE BAD REGISTER ADDRESS
026372 005037 001124          CLR    $GDDAT        ;REGISTER SHOULD BE ZERO
026376 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS THE REGISTER ZERO
026404 001403                    BEQ    64$           ;BR IF IT IS
026406 104004                    EMT    4
026410 000137 027222          JMP    1$           ;BYPASS REST OF THE SUBTEST
026414                    64$:
026414 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
026422 013737 001224 001240  MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026430 016037 000012 001126  MOV    RMDS(RO),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
026436 042737 020001 001126  BIC    #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
026444 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
026452 013737 001124 001166  MOV    $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
026460 005137 001166          COM    $TMP1         ;COMPLEMENT THE EXPECTED STATUS
026464 013737 001126 001164  MOV    $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
026472 043737 001166 001164  BIC    $TMP1,$TMP0   ;CLEAR UNWANTED BITS
026500 023737 001124 001164  CMP    $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
026506 001401                    BEQ    65$           ;BR IF THEY ARE
026510 104005                    EMT    5
026512 000240                    65$:
026514 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
026522 013737 001226 001240  MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;READ RMCS1 THROUGH PORT B - TRY TO SET PORT REQUEST

026530 005037 001250          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
026534 016037 000000 001126  MOV    RMCS1(RO),$BDDAT ;GET CONTENTS OF RMCS1
026542 012737 000000 001122  MOV    #RMCS1,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
026550 060037 001122          ADD    RO,$BDADR     ;ADD RH/RM BASE ADDRESS
026554 005037 001124          CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
026560 013737 001126 001164  MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
026566 042737 173700 001164  BIC    #^C4077,$TMP0 ;SAVE SPECIFIED BITS
026574 023737 001124 001164  CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
026602 001414                    BEQ    66$           ;BR IF OK
026604 013737 001126 001174  MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
026612 042737 004077 001174  BIC    #4077,$TMP4   ;CLEAR THE MASKED BITS
026620 053737 001174 001124  BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
026626 104010                    EMT    10
026630 005137 001250          COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
026634 000240                    66$:
                                NOP

;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT A

026636 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
026644 013737 001224 001240  MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026652 012760 000013 000000  MOV    #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

026660 005037 001254          CLR    RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
026664 012737 000012 001122  MOV    #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
026672 060037 001122          ADD    RO,$BDADR     ;ADD THE I/O BASE ADDRESS
026676 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
026704 113760 001224 000010  MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
    
```



```

026712 016037 000012 001170      MOV      RMDS(R0), $TMP2      ;GET THE DRIVE STATUS REGISTER FROM PORT A.
026720 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2      ;CLEAR DONT CARES
026726 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
026734 042737 100100 001164      BIC      #ATA!VV, $TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026742 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
026750 016037 000012 001172      MOV      RMDS(R0), $TMP3      ;GET THE DRIVE STATUS REGISTER FROM PORT B.
026756 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3      ;CLEAR DONT CARES
026764 013737 001172 001166      MOV      $TMP3, $TMP1      ;COPY IT INTO '$TMP1'
026772 042737 100100 001166      BIC      #ATA!VV, $TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
027000 023737 001164 001166      CMP      $TMP0, $TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
027006 001006      BNE      68$      ;BR IF NOT
027010 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
027014 001045      BNE      70$      ;BR IF NOT
027016 104046      EMT      46
027020 000137 027220      JMP      72$      ;BYPASS THE REST OF THE CHECKS
027024 013737 001170 001126 68$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
027032 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027040 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
027046 005737 001164      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
027052 001414      BEQ      69$      ;BR IF ZERO
027054 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027062 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
027070 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)      ;SELECT PORT A.
027076 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
027102 001012      BNE      70$      ;BR IF NOT
027104 012737 177777 001254 69$:  MOV      #-1, RELERR      ;SET 'RELEASE ERROR' INDICATOR
027112 012760 000011 000000      MOV      #11, RMCS1(R0)      ;CLEAR THE DRIVE
027120 012760 000013 000000      MOV      #13, RMCS1(R0)      ;RELEASE THE DRIVE
027126 104026      EMT      26
027130 013737 001170 001126 70$:  MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
027136 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT NUMBER
027144 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
027152 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;ALL BITS OK ?
027160 001401      BEQ      71$      ;BR IF OK FROM PORT A.
027162 104007      EMT      7
027164 013737 001172 001126 71$:  MOV      $TMP3, $BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
027172 013737 001226 001240      MOV      PORTB, PTNBR      ;CHANGE PORT NUMBER
027200 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
027206 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;SEE IF READ OK FROM PORT B.
027214 001401      BEQ      72$      ;BR IF OK
027216 104007      EMT      7
027220 000240      NOP
027222 000004 1$:  SCOPE      ;LOOP ?
    
```

723  
738  
739

```

*****
*TEST 22      SEIZE BY RMAS TEST
*
*TEST THAT WRITING THE APPROPRIATE DRIVE BIT INTO THE ATTENTION REGISTER
* (RMAS) SEIZES THE DRIVE. VERIFY THAT REQUEST IS SET FOR THE OTHER
* PORT.
*
* A. WRITE THE APPROPRIATE DRIVE BIT INTO RMAS; VERIFY THAT THE DRIVE
* IS SEIZED.
*
* B. ISSUE A RELEASE COMMAND THROUGH THE SEIZING PORT; VERIFY THAT THE
* DRIVE SWITCHES TO THE OPPOSITE PORT. ISSUE A RELEASE THROUGH THE
    
```

OPPOSITE PORT AND VERIFY THAT THE DRIVE IS IN NEUTRAL.

\*\*\*\*\*

027224  
027224 005737 001300  
027230 001406  
027232 100002  
027234 000137 003110  
027240 012737 177777 001300  
027246 012737 027262 001106  
027254 012737 027262 001110  
027262  
027262 112737 000022 001102  
027270 012706 001100  
027274 012737 000031 001176

TST22:  
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?  
BEQ 2\$ ;BR IF NOT  
BPL 1\$ ;BR IF JUST ENTERED TEST  
JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
1\$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR  
2\$: MOV #TEST22,\$LPADR ;SETUP SCOPE LOOP ADDRESS  
MOV #TEST22,\$LPERR ;SETUP ERROR LOOP ADDRESS  
TEST22:  
MOVB #22,\$TSTNM ;MOVE #22 TO TEST NUMBER  
MOV #STACK,\$SP ;LOAD THE STACK POINTER  
MOV #25,\$TIMES ;DO 25. ITERATIONS

740  
794

;CLEAR ATTENTION BITS FOR BOTH PORTS

027302 113760 001224 000010  
027310 005060 000012  
027314 012760 000011 000000  
027322 012760 000013 000000  
027330 113760 001226 000010  
027336 005060 000012  
027342 012760 000011 000000  
027350 012760 000013 000000

MOVB PORTA,RMCS2(R0) ;SELECT PORT #A  
CLR RMDS(R0) ;SEIZE THE DRIVE  
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR  
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE  
MOVB PORTB,RMCS2(R0) ;SELECT PORT #B  
CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'  
MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR  
MOV #13,RMCS1(R0) ;RELEASE THE DRIVE

;SELECT DRIVE OTHER THAN THAT BEING TESTED

027356 113760 001230 000010  
027364 013737 001224 001242

MOVB PORTC,RMCS2(R0) ;SELECT DRIVE NOT BEING TESTED  
MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS

;WRITE THE DRIVE'S ATTENTION BIT

027372 013760 001236 000016  
027400 113760 001224 000010  
027406 013737 001224 001240

MOV ASR1,RMAS(R0) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED  
MOVB PORTA,RMCS2(R0) ;SELECT PORT A  
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;VERIFY THAT EITHER PORT A OR PORT B HAS THE DRIVE

027414 005760 000012  
027420 001014  
027422 113760 001226 000010  
027430 013737 001226 001240  
027436 005760 000012  
027442 001021  
027444 104042  
027446 000137 031276  
027452  
027452 113760 001226 000010  
027460 013737 001226 001240  
027466 005760 000012  
027472 001002  
027474 000137 030376  
027500 104043  
027502 000137 031276

TST RMDS(R0) ;SEE THE REGISTER THROUGH PORT A ?  
BNE 1\$ ;BR IF YES  
MOVB PORTB,RMCS2(R0) ;SELECT PORT B  
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
TST RMDS(R0) ;SEE REGISTER THROUGH PORT B ?  
BNE 2\$ ;BR IF YES  
EMT 42  
JMP 4\$ ;BYPASS REST OF TEST  
1\$: MOVB PORTB,RMCS2(R0) ;SELECT PORT B  
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
TST RMDS(R0) ;REGISTER SHOULD BE ZERO THROUGH PORT B  
BNE .+6 ;BR IF STATUS REG IS NOT ZERO  
JMP 3\$ ;STATUS REG IS ZERO  
EMT 43  
JMP 4\$ ;BYPASS REST OF TEST



;PORT B HAS THE DRIVE. VERIFY THAT PORT A HAS PORT REQUEST SET

```

027506
027506 005037 001250
027512 016037 000012 001126
027520 012737 000012 001122
027526 060037 001122
027532 012737 011700 001124
027540 013737 001126 001164
027546 042737 106077 001164
027554 023737 001124 001164
027562 001414
027564 013737 001126 001174
027572 042737 071700 001174
027600 053737 001174 001124
027606 104010
027610 005137 001250
027614 000240
027616 013737 001226 001242
027624 013737 001224 001244
2$: CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMD5(RO), $BDDAT ;GET CONTENTS OF RMD5
MOV #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #^C71700, $TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT, $TMP0 ;COMPARE THE BITS
BEQ 64$ ;BR IF OK
MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
BIC #71700, $TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
EMT 10
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
64$: NOP
MOV PORTB, SEIZPT ;ADDRESS FOR ERROR MESSAGE
MOV PORTA, OPPRT ;SAME AS ABOVE

```

;RELEASE THE DRIVE FROM PORT B

```

027632 113760 001226 000010
027640 013737 001226 001240
027646 012760 000013 000000
MOV PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

```

;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B

```

027654 005037 001254
027660 012737 111700 001124
027666 012737 000012 001122
027674 060037 001122
027700 113760 001224 000010
027706 013737 001224 001240
027714 016037 000012 001164
027722 113760 001226 000010
027730 013737 001226 001240
027736 016037 000012 001126
027744 001404
027746 005737 001164
027752 001401
027754 104044
027756 013737 001164 001126
027764 013737 001224 001240
027772 023737 001124 001126
030000 001401
030002 104027
030004 000240
CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
MOV #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
MOV #RMD5, $BDADR ;REGISTER ADDRESS INCREMENT
ADD RO, $BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
MOVB PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMD5(RO), $TMP0 ;READ STATUS REGISTER FROM PORT A
MOVB PORTB, RMCS2(RO) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMD5(RO), $BDDAT ;DRIVE STATUS FROM PORT B
BEQ 66$ ;BR IF STATUS FROM PORT B ZERO
TST $TMP0 ;IS STATUS FROM PORT A ZERO?
BEQ 66$ ;BR IF ZERO
EMT 44
66$: MOV $TMP0, $BDDAT ;CHECK STATUS FROM PORT A
MOV PORTA, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
CMP $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
BEQ 67$ ;BR IF OK
EMT 27
67$: NOP

```

;RELEASE THE DRIVE FROM PORT A

```

030006 113760 001224 000010
030014 013737 001224 001240
030022 012760 000013 000000
MOVB PORTA, RMCS2(RO) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

```



;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

030030 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
030034 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
030042 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
030046 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
030054 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
030062 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
030070 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
030076 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
030104 042737 100100 001164  BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030112 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
030120 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
030126 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
030134 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
030142 042737 100100 001166  BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030150 023737 001164 001166  CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
030156 001006          BNE      68$        ;BR IF NOT
030160 005737 001164          TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
030164 001045          BNE      70$        ;BR IF NOT
030166 104046          EMT      46
030170 000137 03C370          JMP      72$
030174 013737 001170 001126 68$:  MOV      $TMP2, $BDDAT ;BYPASS THE REST OF THE CHECKS
030202 013737 001226 001240  MOV      PORTB, PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
030210 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030216 005737 001164          TST      $TMP0      ;SELECT PORT B.
030222 001414          BEQ      69$        ;SEE IF STATUS EQ 0 FROM PORT A.
030224 013737 001224 001240  MOV      PORTA, PTNBR ;BR IF ZERO
030232 013737 001172 001126  MOV      $TMP3, $BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030240 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;'BAD DATA' FOR ERROR TYPE OUT
030246 005737 001166          TST      $TMP1      ;SELECT PORT A.
030252 001012          BNE      70$        ;SEE IF STATUS EQ ZERO FROM PORT B.
030254 012737 177777 001254 69$:  MOV      #-1, RELERR ;BR IF NOT
030262 012760 000011 000000  MOV      #11, RMCS1(RO) ;SET 'RELEASE ERROR' INDICATOR
030270 012760 000013 000000  MOV      #13, RMCS1(RO) ;CLEAR THE DRIVE
030276 104026          EMT      26 ;RELEASE THE DRIVE
030300 013737 001170 001126 70$:  MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
030306 013737 001224 001240  MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
030314 042737 100000 001126  BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
030322 023737 001124 001126  CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
030330 001401          BEQ      71$        ;BR IF OK FROM PORT A.
030332 104007          EMT      7
030334 013737 001172 001126 71$:  MOV      $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
030342 013737 001226 001240  MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
030350 042737 100000 001126  BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
030356 023737 001124 001126  CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
030364 001401          BEQ      72$        ;BR IF OK
030366 104007          EMT      7
030370 000240          NOP
030372 000137 031276          JMP      4$
  
```

;THE DRIVE IS SEIZED BY PORT A. VERIFY THAT PORT B HAS PORT REQUEST SET

```

030376 3$:  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
030376 113760 001224 000010  MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030404 013737 001224 001240  CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
030412 005037 001250
  
```



```

030416 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
030424 012737 000012 001122      MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
030432 060037 001122                ADD      R0, $BDADR     ;ADD RH/RM BASE ADDRESS
030436 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
030444 013737 001126 001164      MOV      $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
030452 042737 106077 001164      BIC      #^C71700, $TMP0 ;SAVE SPECIFIED BITS
030460 023737 001124 001164      CMP      $GDDAT, $TMP0  ;COMPARE THE BITS
030466 001414                BEQ      73$           ;BR IF OK
030470 013737 001126 001174      MOV      $BDDAT, $TMP4  ;COPY 'BAD DATA'
030476 042737 071700 001174      BIC      #71700, $TMP4  ;CLEAR THE MASKED BITS
030504 053737 001174 001124      BIS      $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
030512 104010                EMT      10
030514 005137 001250                COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
030520 000240                73$:  NOP
030522 013737 001224 001242      MOV      PORTA, SEIZPT  ;ADDRESS FOR ERROR MESSAGE
030530 013737 001226 001244      MOV      PORTB, OPPRT   ;SAME AS ABOVE

```

;RELEASE THE DRIVE FROM PORT A

```

030536 113760 001224 000010      MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A
030544 013737 001224 001240      MOV     PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030552 012760 000013 000000      MOV     #13, RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT A

```

;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A

```

030560 005037 001254                CLR      RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
030564 012737 111700 001124      MOV     #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
030572 012737 000012 001122      MOV     #RMDS, $BDADR  ;REGISTER ADDRESS INCREMENT
030600 060037 001122                ADD      R0, $BDADR    ;REGISTER BASE ADDRESS FOR TYPEOUT
030604 113760 001226 000010      MOVVB   PORTB, RMCS2(R0) ;SELECT PORT B
030612 013737 001226 001240      MOV     PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030620 016037 000012 001164      MOV     RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT B
030626 113760 001224 000010      MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A
030634 013737 001224 001240      MOV     PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030642 016037 000012 001126      MOV     RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT A
030650 001404                BEQ      75$          ;BR IF STATUS FROM PORT A ZERO
030652 005737 001164                TST     $TMP0         ;IS STATUS FROM PORT B ZERO ?
030656 001401                BEQ      75$          ;BR IF ZERO
030660 104044                EMT      44
030662 013737 001164 001126      75$:  MOV     $TMP0, $BDDAT ;CHECK STATUS FROM PORT B
030670 013737 001226 001240      MOV     PORTB, PTNBR   ;CHANGE PORT ADDRESS FOR TYPEOUT
030676 023737 001124 001126      CMP     $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
030704 001401                BEQ      76$          ;BR IF OK
030706 104027                EMT      27
030710 000240                76$:  NOP

```

;RELEASE THE DRIVE FROM PORT B

```

030712 113760 001226 000010      MOVVB   PORTB, RMCS2(R0) ;SELECT PORT B
030720 013737 001226 001240      MOV     PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030726 012760 000013 000000      MOV     #13, RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT B

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

030734 005037 001254                CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
030740 012737 000012 001122      MOV     #RMDS, $BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
030746 060037 001122                ADD      R0, $BDADR    ;ADD THE I/O BASE ADDRESS

```

```

030752 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
030760 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
030766 016037 000012 001170      MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
030774 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
031002 013737 001170 001164      MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
031010 042737 100100 001164      BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031016 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
031024 016037 000012 001172      MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
031032 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
031040 013737 001172 001166      MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
031046 042737 100100 001166      BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031054 023737 001164 001166      CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031062 001006      BNE      77$ ;BR IF NOT
031064 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
031070 001045      BNE      79$ ;BR IF NOT
031072 104046      EMT      46
031074 000137 031274      JMP      81$ ;BYPASS THE REST OF THE CHECKS
031100 013737 001170 001126 77$: MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031106 013737 001226 001240      MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031114 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
031122 005737 001164      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
031126 001414      BEQ      78$ ;BR IF ZERO
031130 013737 001224 001240      MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031136 013737 001172 001126      MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
031144 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
031152 005737 001166      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
031156 001012      BNE      79$ ;BR IF NOT
031160 012737 177777 001254 78$: MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
031166 012760 000011 000000      MOV      #11, RMCS1(R0) ;CLEAR THE DRIVE
031174 012760 000013 000000      MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
031202 104026      EMT      26
031204 013737 001170 001126 79$: MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
031212 013737 001224 001240      MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
031220 042737 100000 001126      BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
031226 023737 001124 001126      CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
031234 001401      BEQ      80$ ;BR IF OK FROM PORT A.
031236 104007      EMT      7
031240 013737 001172 001126 80$: MOV      $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
031246 013737 001226 001240      MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
031254 042737 100000 001126      BIC      #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
031262 023737 001124 001126      CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
031270 001401      BEQ      81$ ;BR IF OK
031272 104007      EMT      7
031274 000240      81$: NOP
031276 000004      4$: SCOPE ;LOOP ?

```

795  
807  
808

```

*****
:*TEST 23      INHIBIT SEIZE BY RMAS TEST
:*
:*VERIFY THAT THE DRIVE IS NOT SEIZED WHEN A 'ZERO' IS WRITTEN INTO
:*      THE DRIVE'S ATTENTION BIT.
:*
:*  A.  SELECT A DRIVE NOT BEING TESTED AND WRITE ALL BITS, EXCEPT THE
:*      BIT OF THE DRIVE BEING TESTED, INTO THE ATTENTION REGISTER.
:*
:*  B.  VERIFY THAT THE DRIVE IS STILL IN NEUTRAL.

```





```

031632 013737 001226 001240      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031640 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B.
031646 005737 001164              TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
031652 001414                      BEQ      65$             ;BR IF ZERO
031654 013737 001224 001240      MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031662 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
031670 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A.
031676 005737 001166              TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
031702 001012                      BNE     66$             ;BR IF NOT
031704 012737 177777 001254 65$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
031712 012760 000011 000000      MOV      #11,RMCS1(RO)   ;CLEAR THE DRIVE
031720 012760 000013 000000      MOV      #13,RMCS1(RO)   ;RELEASE THE DRIVE
031726 104021                      EMT      21
031730 013737 001170 001126 66$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
031736 013737 001224 001240      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
031744 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031752 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
031760 001401                      BEQ     67$             ;BR IF OK FROM PORT A.
031762 104007                      EMT      7
031764 013737 001172 001126 67$:  MOV      $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
031772 013737 001226 001240      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
032000 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
032006 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
032014 001401                      BEQ     68$             ;BR IF OK
032016 104007                      EMT      7
032020 000240 68$:  NOP
032022 000004      SCOPE                    ;LOOP ?
    
```

825  
844  
845

```

*****
*TEST 24      SET PORT 'A' REQUEST TEST
*
*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
*DRIVE IS SEIZED BY THE OTHER PORT.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
*
*  B.  WRITE 0'S INTO RMDS FROM PORT 'A'; VERIFY THAT THE DRIVE IS STILL
*      SEIZED BY PORT 'B'.
*
*  C.  ISSUE A RELEASE COMMAND FROM PORT 'B' AND VERIFY THAT THE DRIVE
*      SWITCHED TO PORT 'A'. VERIFY THAT THE ATTENTION BIT IS SET FOR
*      PORT 'A' AND IS NOT SET FOR PORT 'B'.
*
*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'A' AND VERIFY THAT THE DRIVE
*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
    
```

```

032024 005737 001300      TST24:  TST      KYBCTL        ;PERFORMING ONLY SINGLE TEST ?
032024 001406                      BEQ     2$              ;BR IF NOT
032032 100002                      BPL     1$              ;BR IF JUST ENTERED TEST
032034 000137 003110      JMP      EXEC           ;RETURN & GET NEXT TEST NUMBER
032040 012737 177777 001300 1$:  MOV      #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
032046 012737 032062 001106 2$:  MOV      #TEST24,$LPADR ;SETUP SCOPE LOOP ADDRESS
032054 012737 032062 001110      MOV      #TEST24,$LPERR ;SETUP ERROR LOOP ADDRESS
032062
    
```



846  
875

```

032062 112737 000024 001102      MOVB  #24,$STNM      ;MOVE #24 TO TEST NUMBER
032070 012706 001100              MOV   #STACK,SP     ;LOAD THE STACK POINTER
032074 012737 000031 001176      MOV   #25.,$TIMES   ;;DO 25. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

032102 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
032110 005060 000012              CLR   RMDS(R0)       ;SEIZE THE DRIVE
032114 012760 000011 000000      MOV   #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
032122 012760 000013 000000      MOV   #13, RMCS1(R0) ;RELEASE THE DRIVE
032130 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
032136 005060 000012              CLR   RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
032142 012760 000011 000000      MOV   #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
032150 012760 000013 000000      MOV   #13, RMCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

032156 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
032164 013737 001226 001242      MOV   PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
032172 005060 000012              CLR   RMDS(R0)       ;WRITE RMDS
032176 013737 001224 001244      MOV   PORTA, OPPRT   ;'OPPOSITE' PORT ADDRESS
032204 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
032212 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET PORT REQUEST

032220 005060 000012              CLR   RMDS(R0)       ;SET PORT REQUEST FOR PORT A

;RELEASE THROUGH PORT B. DRIVE SHOULD SWITCH TO PORT A.

;RELEASE THE DRIVE FROM PORT B

032224 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
032232 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032240 012760 000013 000000      MOV   #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B

032246 005037 001254              CLR   RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
032252 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
032260 012737 000012 001122      MOV   #RMDS,$BDADR   ;REGISTER ADDRESS INCREMENT
032266 060037 001122              ADD   R0,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
032272 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
032300 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032306 016037 000012 001164      MOV   RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT A
032314 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
032322 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032330 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT B
032336 001404              BEQ   66$           ;BR IF STATUS FROM PORT B ZERO
032340 005737 001164              TST   $TMP0         ;IS STATUS FROM PORT A ZERO ?
032344 001401              BEQ   66$           ;BR IF ZERO
032346 104031              EMT   31
032350 013737 001164 001126      66$: MOV   $TMP0,$BDDAT   ;CHECK STATUS FROM PORT A
032356 013737 001224 001240      MOV   PORTA, PTNBR  ;CHANGE PORT ADDRESS FOR TYPEOUT
032364 023737 001124 001126      CMP   $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
032372 001401              BEQ   67$           ;BR IF OK
    
```

```

032374 104027          EMT      27
032376 000240          NOP
032400 113760 001226 000010 67$:  MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
032406 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032414 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
032420 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
032426 012737 000012 001122      MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032434 060037 001122          ADD   R0, $BDADR ;ADD RH/RM BASE ADDRESS
032440 005037 001124          CLR   $GDDAT ;WHAT REGISTER SHOULD BE
032444 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
032452 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
032460 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
032466 001414          BEQ   68$ ;BR IF OK
032470 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
032476 042737 100000 001174      BIC   #ATA, $TMP4 ;CLEAR THE MASKED BITS
032504 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
032512 104016          EMT      16
032514 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
032520 000240          NOP
032522 113760 001224 000010 68$:  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
032530 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032536 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
032542 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
032550 012737 000012 001122      MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032556 060037 001122          ADD   R0, $BDADR ;ADD RH/RM BASE ADDRESS
032562 012737 100000 001124      MOV   #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
032570 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
032576 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
032604 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
032612 001414          BEQ   70$ ;BR IF OK
032614 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
032622 042737 100000 001174      BIC   #ATA, $TMP4 ;CLEAR THE MASKED BITS
032630 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
032636 104016          EMT      16
032640 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
032644 000240          NOP
    
```

;RELEASE THE DRIVE FROM PORT A

```

032646 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
032654 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032662 012760 000013 000000      MOV   #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

032670 005037 001254          CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
032674 012737 000012 001122      MOV   #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
032702 060037 001122          ADD   R0, $BDADR ;ADD THE I/O BASE ADDRESS
032706 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
032714 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
032722 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
032730 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
032736 013737 001170 001164      MOV   $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
032744 042737 100100 001164      BIC   #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032752 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
032760 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
032766 042737 024001 001172      BIC   #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
    
```



```

032774 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
033002 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
033010 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
033016 001006                BNE      72$              ;BR IF NOT
033020 005737 001164                TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
033024 001045                BNE      74$              ;BR IF NOT
033026 104046                EMT      46
033030 000137 033214                JMP      76$              ;BYPASS THE REST OF THE CHECKS
033034 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
033042 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033050 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B.
033056 005737 001164                TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
033062 001414                BEQ      73$              ;BR IF ZERO
033064 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033072 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
033100 113760 001224 000010      MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
033106 005737 001166                TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
033112 001012                BNE      74$              ;BR IF NOT
033114 012737 177777 001254 73$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
033122 012760 000011 000000      MOV      #11,RMCS1(RO)   ;CLEAR THE DRIVE
033130 012760 000013 000000      MOV      #13,RMCS1(RO)   ;RELEASE THE DRIVE
033136 104026                EMT      26
033140 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
033146 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
033154 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
033162 001401                BEQ      75$              ;BR IF OK FROM PORT A.
033164 104007                EMT      7
033166 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
033174 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
033202 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
033210 001401                BEQ      76$              ;BR IF OK
033212 104007                EMT      7
033214 000240                NOP
033216 000004                1$:      SCOPE              ;LOOP ?
    
```

876  
895  
896

```

*****
*TEST 25      SET PORT 'B' REQUEST TEST
*
*VERIFY THAT WRITING A DRIVE REGISTER SETS 'PORT REQUEST' WHEN THE
*DRIVE IS SEIZED BY THE OTHER PORT.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*
*  B.  WRITE 0'S INTO RMDS FROM PORT 'B'; VERIFY THAT THE DRIVE IS STILL
*      SEIZED BY PORT 'A'.
*
*  C.  ISSUE A RELEASE COMMAND FROM PORT 'A' AND VERIFY THAT THE DRIVE
*      SWITCHED TO PORT 'B'. VERIFY THAT THE ATTENTION BIT IS SET FOR
*      PORT 'B' AND IS NOT SET FOR PORT 'A'.
*
*  D.  ISSUE A RELEASE COMMAND THROUGH PORT 'B' AND VERIFY THAT THE DRIVE
*      RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
TST25:      TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
    
```

033220  
033220 005737 001300

897  
898

```

033224 001406          BEQ      2$          ;BR IF NOT
033226 100002          BPL      1$          ;BR IF JUST ENTERED TEST
033230 000137 003110   JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
033234 012737 177777 001300 1$: MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
033242 012737 033256 001106 2$: MOV      #TEST25,$LPADR ;SETUP SCOPE LOOP ADDRESS
033250 012737 033256 001110   MOV      #TEST25,$LPERR ;SETUP ERROR LOOP ADDRESS
033256                                TEST25:
033256 112737 000025 001102   MOVB     #25,$TSTNM   ;MOVE #25 TO TEST NUMBER
033264 012706 001100          MOV      #STACK,SP   ;LOAD THE STACK POINTER
033270 012737 000031 001176   MOV      #25.,$TIMES ;DO 25. ITERATIONS
  
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

033276 113760 001224 000010   MOVB     PORTA, RMCS2(R0) ;SELECT PORT #A
033304 005060 000012          CLR      RMDS(R0)      ;SEIZE THE DRIVE
033310 012760 000011 000000   MOV      #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
033316 012760 000013 000000   MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
033324 113760 001226 000010   MOVB     PORTB, RMCS2(R0) ;SELECT PORT #B
033332 005060 000012          CLR      RMDS(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
033336 012760 000011 000000   MOV      #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
033344 012760 000013 000000   MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

033352 113760 001224 000010   MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
033360 013737 001224 001242   MOV      PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
033366 005060 000012          CLR      RMDS(R0)      ;WRITE RMDS
033372 013737 001226 001244   MOV      PORTB, OPPRT  ;'OPPOSITE' PORT ADDRESS
033400 113760 001226 000010   MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
033406 013737 001226 001240   MOV      PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

;SET PORT REQUEST

```

033414 005060 000012          CLR      RMDS(R0)      ;SET PORT REQUEST FOR PORT B
  
```

;RELEASE THROUGH PORT A. DRIVE SHOULD SWITCH TO PORT B.

;RELEASE THE DRIVE FROM PORT A

```

033420 113760 001224 000010   MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
033426 013737 001224 001240   MOV      PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033434 012760 000013 000000   MOV      #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A

```

033442 005037 001254          CLR      RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
033446 012737 111700 001124   MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
033454 012737 000012 001122   MOV      #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
033462 060037 001122          ADD      R0,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
033466 113760 001226 000010   MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
033474 013737 001226 001240   MOV      PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033502 016037 000012 001164   MOV      RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT B
033510 113760 001224 000010   MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
033516 013737 001224 001240   MOV      PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033524 016037 000012 001126   MOV      RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT A
033532 001404          BEQ      66$          ;BR IF STATUS FROM PORT A ZERO
  
```



```

033534 005737 001164          TST      $TMP0          ;IS STATUS FROM PORT B ZERO ?
033540 001401          BEQ      66$          ;BR IF ZERO
033542 104031          EMT      31
033544 013737 001164 001126 66$:  MOV     $TMP0,$BDDAT  ;CHECK STATUS FROM PORT B
033552 013737 001226 001240  MOV     PORTB,PTNBR  ;CHANGE PORT ADDRESS FOR TYPEOUT
033560 023737 001124 001126  CMP     $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
033566 001401          BEQ      67$          ;BR IF OK
033570 104027          EMT      27
033572 000240          NOP
033574 113760 001224 000010 67$:  MOVVB  PORTA,RMCS2(R0) ;SELECT PORT A
033602 013737 001224 001240  MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033610 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
033614 016037 000012 001126  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
033622 012737 000012 001122  MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033630 060037 001122          ADD     R0,$BDADR    ;ADD RH/RM BASE ADDRESS
033634 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
033640 013737 001126 001164  MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
033646 042737 077777 001164  BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
033654 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
033662 001414          BEQ      68$          ;BR IF OK
033664 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
033672 042737 100000 001174  BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
033700 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
033706 104016          EMT      16
033710 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
033714 000240          NOP
033716 113760 001226 000010 68$:  MOVVB  PORTB,RMCS2(R0) ;SELECT PORT B
033724 013737 001226 001240  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033732 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
033736 016037 000012 001126  MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
033744 012737 000012 001122  MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033752 060037 001122          ADD     R0,$BDADR    ;ADD RH/RM BASE ADDRESS
033756 012737 100000 001124  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
033764 013737 001126 001164  MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
033772 042737 077777 001164  BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
034000 023737 001124 001164  CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
034006 001414          BEQ      70$          ;BR IF OK
034010 013737 001126 001174  MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
034016 042737 100000 001174  BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
034024 053737 001174 001124  BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
034032 104016          EMT      16
034034 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
034040 000240          NOP
    
```

;RELEASE THE DRIVE FROM PORT B

```

034042 113760 001226 000010  MOVVB  PORTB,RMCS2(R0) ;SELECT PORT B
034050 013737 001226 001240  MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034056 012760 000013 000000  MOV     #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

034064 005037 001254          CLR     RELERR       ;CLEAR THE 'RELEASE ERROR ' INDICATOR
034070 012737 000012 001122  MOV     #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
034076 060037 001122          ADD     R0,$BDADR    ;ADD THE I/O BASE ADDRESS
034102 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
034110 113760 001224 000010  MOVVB  PORTA,RMCS2(R0) ;SELECT PORT A.
    
```

```

034116 016037 000012 001170      MOV      RMDS(RO), $TMP2      ;GET THE DRIVE STATUS REGISTER FROM PORT A.
034124 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2      ;CLEAR DONT CARES
034132 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
034140 042737 100100 001164      BIC      #ATA!VV, $TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034146 113760 001226 000010      MOV      PORTB, RMCS2(RO)      ;SELECT PORT B.
034154 016037 000012 001172      MOV      RMDS(RO), $TMP3      ;GET THE DRIVE STATUS REGISTER FROM PORT B.
034162 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3      ;CLEAR DONT CARES
034170 013737 001172 001166      MOV      $TMP3, $TMP1      ;COPY IT INTO '$TMP1'
034176 042737 100100 001166      BIC      #ATA!VV, $TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034204 023737 001164 001166      CMP      $TMP0, $TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
034212 001006      BNE      72$      ;BR IF NOT
034214 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
034220 001045      BNE      74$      ;BR IF NOT
034222 104046      EMT      46
034224 000137 034410      JMP      76$      ;BYPASS THE REST OF THE CHECKS
034230 013737 001170 001126 72$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
034236 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034244 113760 001226 000010      MOV      PORTB, RMCS2(RO)      ;SELECT PORT B.
034252 005737 001164      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
034256 001414      BEQ      73$      ;BR IF ZERO
034260 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034266 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
034274 113760 001224 000010      MOV      PORTA, RMCS2(RO)      ;SELECT PORT A.
034302 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
034306 001012      BNE      74$      ;BR IF NOT
034310 012737 177777 001254 73$:  MOV      #-1, RELERR      ;SET 'RELEASE ERROR' INDICATOR
034316 012760 000011 000000      MOV      #11, RMCS1(RO)      ;CLEAR THE DRIVE
034324 012760 000013 000000      MOV      #13, RMCS1(RO)      ;RELEASE THE DRIVE
034332 104026      EMT      26
034334 013737 001170 001126 74$:  MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
034342 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT NUMBER
034350 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;ALL BITS OK ?
034356 001401      BEQ      75$      ;BR IF OK FROM PORT A.
034360 104007      EMT      7
034362 013737 001172 001126 75$:  MOV      $TMP3, $BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
034370 013737 001226 001240      MOV      PORTB, PTNBR      ;CHANGE PORT NUMBER
034376 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;SEE IF READ OK FROM PORT B.
034404 001401      BEQ      76$      ;BR IF OK
034406 104007      EMT      7
034410 000240      NOP
034412 000004 76$:  1$:  SCOPE      ;LOOP ?
  
```

902  
921  
922

```

*****
*TEST 26      TEST RESET ATTENTION 'A' BY DRIVE CLEAR
*
*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
*SEIZING PORT.
*
* A. SET EACH PORT 'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
*SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*
* C. ISSUE A DRIVE CLEAR COMMAND.
*
*****
  
```



;\* D. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE ATTENTION  
 BIT FOR PORT 'A' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT  
 'B' IS STILL SET.  
 ;\*

\*\*\*\*\*

034414  
 034414 005737 001300  
 034420 001406  
 034422 100002  
 034424 000137 003110  
 034430 012737 177777 001300  
 034436 012737 034452 001106  
 034444 012737 034452 001110  
 034452  
 034452 112737 000026 001102  
 034460 012706 001100  
 034464 012737 000031 001176  
 923  
 956

TST26:  
 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?  
 BEQ 2\$ ;BR IF NOT  
 BPL 1\$ ;BR IF JUST ENTERED TEST  
 JMP EXEC ;RETURN & GET NEXT TEST NUMBER  
 1\$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR  
 2\$: MOV #TEST26,\$LPADR ;SETUP SCOPE LOOP ADDRESS  
 MOV #TEST26,\$LPERR ;SETUP ERROR LOOP ADDRESS  
 TEST26:  
 MOVB #26,\$TSTNM ;MOVE #26 TO TEST NUMBER  
 MOV #STACK,\$SP ;LOAD THE STACK POINTER  
 MOV #25,\$TIMES ;DO 25. ITERATIONS

;SET ATTENTION BITS FOR BOTH PORTS

034472 113760 001224 000010  
 034500 005760 000012  
 034504 001775  
 034506 012760 177777 000014  
 034514 005060 000014  
 034520 013760 001226 000010  
 034526 005760 000012  
 034532 001775  
 034534 012760 177777 000014  
 034542 005060 000014  
 034546 113760 001224 000010  
 034554 005760 000012  
 034560 001775

66\$: MOVB PORTA,RMCS2(R0) ;SELECT PORT 64\$  
 TST RMD5(R0) ;MAKE SURE DRIVE AVAILABLE  
 BEQ 66\$  
 MOV #-1,RMER1(R0) ;FORCE ERRORS  
 CLR RMER1(R0) ;CLEAR THE ERRORS  
 64\$: MOV PORTB,RMCS2(R0) ;SELECT THE OTHER PORT  
 TST RMD5(R0) ;WAIT FOR DRIVE TO TIMEOUT  
 BEQ 64\$ ;BR IF DRIVE HASN'T TIMED OUT  
 MOV #-1,RMER1(R0) ;FORCE ERRORS ON PORT 65\$  
 CLR RMER1(R0) ;CLEAR THE ERRORS  
 65\$: MOVB PORTA,RMCS2(R0) ;SELECT PORT '64\$' AGAIN  
 TST RMD5(R0) ;WAIT FOR DRIVE TO TIMEOUT  
 BEQ 65\$ ;BR IF DRIVE HASN'T TIMED OUT

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

034562 113760 001224 000010  
 034570 013737 001224 001240  
 034576 005037 001250  
 034602 016037 000012 001126  
 034610 012737 000012 001122  
 034616 060037 001122  
 034622 012737 100000 001124  
 034630 013737 001126 001164  
 034636 042737 077777 001164  
 034644 023737 001124 001164  
 034652 001414  
 034654 013737 001126 001174  
 034662 042737 100000 001174  
 034670 053737 001174 001124  
 034676 104010  
 034700 005137 001250  
 034704 000240  
 034706 005737 001250  
 034712 001402  
 034714 000137 036106

MOVB PORTA,RMCS2(R0) ;SELECT PORT A  
 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT  
 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR  
 MOV RMD5(R0),\$BDDAT ;GET CONTENTS OF RMD5  
 MOV #RMD5,\$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE  
 ADD R0,\$BDADR ;ADD RH/RM BASE ADDRESS  
 MOV #ATA,\$GDDAT ;WHAT REGISTER SHOULD BE  
 MOV \$BDDAT,\$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'  
 BIC #^CATA,\$TMP0 ;SAVE SPECIFIED BITS  
 CMP \$GDDAT,\$TMP0 ;COMPARE THE BITS  
 BEQ 67\$ ;BR IF OK  
 MOV \$BDDAT,\$TMP4 ;COPY 'BAD DATA'  
 BIC #ATA,\$TMP4 ;CLEAR THE MASKED BITS  
 BIS \$TMP4,\$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT  
 EMT 10  
 67\$: COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR  
 NOP  
 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?  
 BEQ +6 ;BR IF IT WAS  
 JMP 1\$ ;BYPASS REST OF TEST IF NOT

```

034720 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
034726 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034734 005037 001250              CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
034740 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
034746 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034754 060037 001122              ADD   RO, $BDADR    ;ADD RH/RM BASE ADDRESS
034760 012737 100000 001124      MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
034766 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
034774 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
035002 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
035010 001414              BEQ   69$          ;BR IF OK
035012 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
035020 042737 100000 001174      BIC   #ATA, $TMP4  ;CLEAR THE MASKED BITS
035026 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
035034 104010              EMT   10
035036 005137 001250              COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
035042 000240              69$: NOP
035044 005737 001250              TST   CKERR        ;WAS ATTN BIT FOR PORT B SET ?
035050 001402              BEQ   .+6          ;BR IF IT WAS
035052 000137 036106              JMP   1$          ;BYPASS REST OF TEST IF NOT
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

035056 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
035064 013737 001224 001242      MOV   PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
035072 005060 000012              CLR   RMDS(R0)     ;WRITE RMDS
035076 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
035104 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035112 013737 001226 001244      MOV   PORTB, OPPRT  ;'OPPOSITE' PORT ADDRESS
035120 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
035126 010037 001122              MOV   RO, $BDADR    ;RH/RM BASE ADDRESS
035132 062737 000012 001122      ADD   #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
035140 005037 001124              CLR   $GDDAT       ;REGISTER SHOULD BE ZERO
035144 023737 001124 001126      CMP   $GDDAT, $BDDAT ;IS THE REGISTER ZERO
035152 001403              BEQ   71$          ;BR IF IT IS
035154 104004              EMT   4
035156 000137 036106              JMP   1$          ;BYPASS REST OF THE SUBTEST
035162 000000              71$:
  
```

```

035162 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
035170 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035176 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
035204 042737 020001 001126      BIC   #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
035212 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
035220 013737 001124 001166      MOV   $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
035226 005137 001166              COM   $TMP1        ;COMPLEMENT THE EXPECTED STATUS
035232 013737 001126 001164      MOV   $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
035240 043737 001166 001164      BIC   $TMP1, $TMP0 ;CLEAR UNWANTED BITS
035246 023737 001124 001164      CMP   $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
035254 001401              BEQ   72$          ;BR IF THEY ARE
035256 104005              EMT   5
035260 000240              72$: NOP
  
```

;ISSUE DRIVE CLEAR COMMAND TO PORT A

```

035262 012760 000011 000000      MOV   #11, RMCS1(R0) ;DO A DRIVE CLEAR COMMAND
  
```

;VERIFY THAT ATTENTION BIT FOR PORT A CLEARED



```

035270 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
035274 016037 000012 001126  MOV      RMD5($BDDAT) ;GET CONTENTS OF RMD5
035302 012737 000012 001122  MOV      #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
035310 060037 001122          ADD      RO,$BDADR     ;ADD RH/RM BASE ADDRESS
035314 005037 001124          CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
035320 013737 001126 001164  MOV      $BDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
035326 042737 077777 001164  BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
035334 023737 001124 001164  CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
035342 001414          BEQ      73$          ;BR IF OK
035344 013737 001126 001174  MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
035352 042737 100000 001174  BIC     #ATA,$TMP4    ;CLEAR THE MASKED BITS
035360 053737 001174 001124  BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
035366 104047          EMT      47
035370 005137 001250          COM      CKERR
035374 000240          NOP

73$:
;RELEASE THE DRIVE FROM PORT A

035376 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A
035404 013737 001224 001240  MOV     PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035412 012760 000013 000000  MOV     #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

035420 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
035424 012737 000012 001122  MOV      #RMD5,$BDADR ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
035432 060037 001122          ADD      RO,$BDADR     ;ADD THE I/O BASE ADDRESS
035436 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
035444 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
035452 016037 000012 001170  MOV      RMD5(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
035460 042737 024001 001170  BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
035466 013737 001170 001164  MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
035474 042737 100100 001164  BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035502 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
035510 016037 000012 001172  MOV      RMD5(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
035516 042737 024001 001172  BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
035524 013737 001172 001166  MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
035532 042737 100100 001166  BIC     #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035540 023737 001164 001166  CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
035546 001006          BNE     75$          ;BR IF NOT
035550 005737 001164          TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
035554 001045          BNE     77$          ;BR IF NOT
035556 104046          EMT      46
035560 000137 035760          JMP     79$          ;BYPASS THE REST OF THE CHECKS
035564 013737 001170 001126 75$: MOV     $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
035572 013737 001226 001240  MOV     PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035600 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
035606 005737 001164          TST     $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
035612 001414          BEQ     76$          ;BR IF ZERO
035614 013737 001224 001240  MOV     PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035622 013737 001172 001126  MOV     $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
035630 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
035636 005737 001166          TST     $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
035642 001012          BNE     77$          ;BR IF NOT
035644 012737 177777 001254 76$: MOV     #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
035652 012760 000011 000000  MOV     #11, RMCS1(RO) ;CLEAR THE DRIVE
  
```



```

035660 012760 000013 000000      MOV    #13,RMCS1(R0)  ;RELEASE THE DRIVE
035666 104026      EMT    26
035670 013737 001170 001126 77$:  MOV    $TMP2,$BDDAT  ;LOOK FOR BIT FAILURES WHEN RMDS READ
035676 013737 001224 001240      MOV    PORTA,PTNBR   ;CHANGE PORT NUMBER
035704 042737 100000 001126      BIC    #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
035712 023737 001124 001126      CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
035720 001401      BEQ    78$          ;BR IF OK FROM PORT A.
035722 104007      EMT    7
035724 013737 001172 001126 78$:  MOV    $TMP3,$BDDAT  ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
035732 013737 001226 001240      MOV    PORTB,PTNBR  ;CHANGE PORT NUMBER
035740 042737 100000 001126      BIC    #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
035746 023737 001124 001126      CMP    $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
035754 001401      BEQ    79$          ;BR IF OK
035756 104007      EMT    7
035760 000240      79$:  NOP
  
```

;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)

```

035762 113760 001226 000010      MOV    PORTB,RMCS2(R0) ;SELECT PORT B
035770 013737 001226 001240      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035776 005037 001250      CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
036002 016037 000012 001126      MOV    RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
036010 012737 000012 001122      MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036016 060037 001122      ADD    R0,$BDADR     ;ADD RH/RM BASE ADDRESS
036022 012737 100000 001124      MOV    #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
036030 013737 001126 001164      MOV    $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
036036 042737 077777 001164      BIC    #^CATA,$TMP0  ;SAVE SPECIFIED BITS
036044 023737 001124 001164      CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
036052 001414      BEQ    80$          ;BR IF OK
036054 013737 001126 001174      MOV    $BDDAT,$TMP4  ;COPY 'BAD DATA'
036062 042737 100000 001174      BIC    #ATA,$TMP4    ;CLEAR THE MASKED BITS
036070 053737 001174 001124      BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
036076 104050      EMT    50
036100 005137 001250      COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
036104 000240      80$:  NOP
036106 000004      1$:  SCOPE          ;LOOP ?
  
```

957  
975  
976

```

*****
*TEST 27      TEST RESET ATTENTION 'B' BY DRIVE CLEAR
*
*VERIFY THAT A DRIVE CLEAR COMMAND CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A. SET EACH PORT'S ATTENTION BIT. VERIFY THAT BOTH ATTENTION BITS
* SET.
*
* B. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
*
* C. ISSUE A DRIVE CLEAR COMMAND.
*
* D. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE ATTENTION
* BIT FOR PORT 'B' HAS BEEN CLEARED AND THE ATTENTION BIT FOR PORT
* 'A' IS STILL SET.
*****
  
```

036110

TST27:



977  
978

```

036110 005737 001300          TST      KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
036114 001406                   BEQ      2$              ;BR IF NOT
036116 100002                   BPL      1$              ;BR IF JUST ENTERED TEST
036120 000137 003110          JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
036124 012737 177777 001300 1$:  MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
036132 012737 036146 001106 2$:  MOV      #TEST27,$LPADR ;SETUP SCOPE LOOP ADDRESS
036140 012737 036146 001110    MOV      #TEST27,$LPERR ;SETUP ERROR LOOP ADDRESS
036146                                TEST27:
036146 112737 000027 001102    MOVB     #27,$TSTNM      ;MOVE #27 TO TEST NUMBER
036154 012706 001100          MOV      #STACK,SP      ;LOAD THE STACK POINTER
036160 012737 000031 001176    MOV      #25.,$TIMES    ;;DO 25. ITERATIONS
  
```

;SET ATTENTION BITS FOR BOTH PORTS

```

036166 113760 001224 000010    66$:  MOVB     PORTA, RMCS2(R0) ;SELECT PORT 64$
036174 005760 000012          TST      RMDS(R0)       ;MAKE SURE DRIVE AVAILABLE
036200 001775                   BEQ      66$
036202 012760 177777 000014    MOV      #-1, RMER1(R0) ;FORCE ERRORS
036210 005060 000014          CLR      RMER1(R0)      ;CLEAR THE ERRORS
036214 013760 001226 000010    MOV      PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
036222 005760 000012    64$:  TST      RMDS(R0)       ;WAIT FOR DRIVE TO TIMEOUT
036226 001775                   BEQ      64$           ;BR IF DRIVE HASN'T TIMED OUT
036230 012760 177777 000014    MOV      #-1, RMER1(R0) ;FORCE ERRORS ON PORT 65$
036236 005060 000014          CLR      RMER1(R0)      ;CLEAR THE ERRORS
036242 113760 001224 000010    MOVB     PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
036250 005760 000012    65$:  TST      RMDS(R0)       ;WAIT FOR DRIVE TO TIMEOUT
036254 001775                   BEQ      65$           ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

036256 113760 001226 000010    MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
036264 013737 001226 001240    MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036272 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
036276 016037 000012 001126    MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036304 012737 000012 001122    MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036312 060037 001122          ADD      RO, $BDADR      ;ADD RH/RM BASE ADDRESS
036316 012737 100000 001124    MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
036324 013737 001126 001164    MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
036332 042737 077777 001164    BIC      #^CATA, $TMP0   ;SAVE SPECIFIED BITS
036340 023737 001124 001164    CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
036346 001414                   BEQ      67$           ;BR IF OK
036350 013737 001126 001174    MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
036356 042737 100000 001174    BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
036364 053737 001174 001124    BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
036372 104010                   EMT      10
036374 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
036400 000240    67$:  NOP
036402 005737 001250          TST      CKERR          ;WAS ATTN BIT FOR PORT B SET ?
036406 001402                   BEQ      .+6           ;BR IF IT WAS
036410 000137 037602          JMP      1$            ;BYPASS REST OF TEST IF NOT
036414 113760 001224 000010    MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
036422 013737 001224 001240    MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036430 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
036434 016037 000012 001126    MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036442 012737 000012 001122    MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036450 060037 001122          ADD      RO, $BDADR      ;ADD RH/RM BASE ADDRESS
  
```

```

036454 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
036462 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
036470 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
036476 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
036504 001414                      BEQ      69$          ;BR IF OK
036506 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
036514 042737 100000 001174      BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
036522 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
036530 104010                      EMT      10
036532 005137 001250                      COM      CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
036536 000240                      69$:  NOP
036540 005737 001250                      TST      CKERR        ;WAS ATTN BIT FOR PORT A SET ?
036544 001402                      BEQ      .+6          ;BR IF IT WAS
036546 000137 037602                      JMP      1$          ;BYPASS REST OF TEST IF NOT
    
```

;SEIZE THE DRIVE THROUGH PORT B

```

036552 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B
036560 013737 001226 001242      MOV      PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
036566 005060 000012                      CLR      RMDS(R0)      ;WRITE RMDS
036572 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A
036600 013737 001224 001240      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036606 013737 001224 001244      MOV      PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
036614 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
036622 010037 001122                      MOV      R0, $BDADR   ;RH/RM BASE ADDRESS
036626 062737 000012 001122      ADD      #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
036634 005037 001124                      CLR      $GDDAT       ;REGISTER SHOULD BE ZERO
036640 023737 001124 001126      CMP      $GDDAT, $BDDAT ;IS THE REGISTER ZERO
036646 001403                      BEQ      71$          ;BR IF IT IS
036650 104004                      EMT      4
036652 000137 037602                      JMP      1$          ;BYPASS REST OF THE SUBTEST
036656
    
```

71\$:

```

036656 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B
036664 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036672 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
036700 042737 020001 001126      BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
036706 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
036714 013737 001124 001166      MOV      $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
036722 005137 001166                      COM      $TMP1        ;COMPLEMENT THE EXPECTED STATUS
036726 013737 001126 001164      MOV      $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
036734 043737 001166 001164      BIC      $TMP1, $TMP0 ;CLEAR UNWANTED BITS
036742 023737 001124 001164      CMP      $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
036750 001401                      BEQ      72$          ;BR IF THEY ARE
036752 104005                      EMT      5
036754 000240                      72$:  NOP
    
```

72\$:

;ISSUE DRIVE CLEAR COMMAND TO PORT B

```

036756 012760 000011 000000      MOV      #11, RMCS1(R0) ;DO A DRIVE CLEAR COMMAND
    
```

;VERIFY THAT ATTENTION BIT FOR PORT B CLEARED

```

036764 005037 001250                      CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
036770 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036776 012737 000012 001122      MOV      #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037004 060037 001122                      ADD      R0, $BDADR   ;ADD RH/RM BASE ADDRESS
037010 005037 001124                      CLR      $GDDAT       ;WHAT REGISTER SHOULD BE
    
```



```

037014 013737 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
037022 042737 077777 001164      BIC      #^CATA,$TMP0     ;SAVE SPECIFIED BITS
037030 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
037036 001414                      BEQ      73$              ;BR IF OK
037040 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
037046 042737 100000 001174      BIC      #ATA,$TMP4       ;CLEAR THE MASKED BITS
037054 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
037062 104047                      EMT      47
037064 005137 001250                      COM      CKERR
037070 000240                      NOP

73$:
;RELEASE THE DRIVE FROM PORT B

037072 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B
037100 013737 001226 001240      MOV      PORTB, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037106 012760 000013 000000      MOV      #13, RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

037114 005037 001254                      CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
037120 012737 000012 001122      MOV      #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
037126 060037 001122                      ADD      R0,$BDADR        ;ADD THE I/O BASE ADDRESS
037132 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
037140 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
037146 016037 000012 001170      MOV      RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
037154 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
037162 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
037170 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037176 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
037204 016037 000012 001172      MOV      RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
037212 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
037220 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
037226 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037234 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
037242 001006                      BNE      75$              ;BR IF NOT
037244 005737 001164                      TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
037250 001045                      BNE      77$              ;BR IF NOT
037252 104046                      EMT      46
037254 000137 037454                      JMP      79$              ;BYPASS THE REST OF THE CHECKS
037260 013737 001170 001126 75$: MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
037266 013737 001226 001240      MOV      PORTB, PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037274 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
037302 005737 001164                      TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
037306 001414                      BEQ      76$              ;BR IF ZERO
037310 013737 001224 001240      MOV      PORTA, PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037316 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
037324 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
037332 005737 001166                      TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
037336 001012                      BNE      77$              ;BR IF NOT
037340 012737 177777 001254 76$: MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
037346 012760 000011 000000      MOV      #11, RMCS1(R0)   ;CLEAR THE DRIVE
037354 012760 000013 000000      MOV      #13, RMCS1(R0)   ;RELEASE THE DRIVE
037362 104026                      EMT      26
037364 013737 001170 001126 77$: MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
037372 013737 001224 001240      MOV      PORTA, PTNBR     ;CHANGE PORT NUMBER
037400 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
037406 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?

```

```

037414 001401          BEQ      78$          ;BR IF OK FROM PORT A.
037416 104007          EMT      7
037420 013737 001172 001126 78$:  MOV     $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
037426 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
037434 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
037442 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
037450 001401          BEQ      79$          ;BR IF OK
037452 104007          EMT      7
037454 000240          79$:  NOP
    
```

;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT A)

```

037456 113760 001224 000010      MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A
037464 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037472 005037 001250              CLR     CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
037476 016037 000012 001126      MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
037504 012737 000012 001122      MOV     #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037512 060037 001122              ADD     R0,$BDADR ;ADD RH/RM BASE ADDRESS
037516 012737 100000 001124      MOV     #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
037524 013737 001126 001164      MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
037532 042737 077777 001164      BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
037540 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
037546 001414          BEQ      80$          ;BR IF OK
037550 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
037556 042737 100000 001174      BIC     #ATA,$TMP4 ;CLEAR THE MASKED BITS
037564 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
037572 104050          EMT      50
037574 005137 001250          COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
037600 000240          80$:  NOP
037602 000004          1$:  SCOPE ;LOOP ?
    
```

979  
998  
999

```

*****
*TEST 30      RESET ATTENTION 'A' BY GO TEST
*
* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A.  SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
* ATTENTION BITS ARE SET.
*
* B.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S
* INTO RMDS.
*
* C.  ISSUE A NOP COMMAND.
*
* D.  RELEASE THE FRIVE THROUGH PORT 'A'. VERIFY THAT THE
* ATTENTION BIT FOR PORT 'A' IS RESET, AND THE
* ATTENTION BIT FOR PORT 'B' IS STIL SET.
*****
    
```

```

037604          TST30:
037604 005737 001300      TST     KYBCTL ;PERFORMING ONLY SINGLE TEST ?
037610 001406          BEQ     2$      ;BR IF NOT
037612 100002          BPL     1$      ;BR IF JUST ENTERED TEST
037614 000137 003110      JMP     EXEC ;RETURN & GET NEXT TEST NUMBER
037620 012737 177777 001300 1$:  MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
    
```



1000  
1033

```

037626 012737 037642 001106 2$: MOV #TEST30,$LPADR ;SETUP SCOPE LOOP ADDRESS
037634 012737 037642 001110 MOV #TEST30,$LPERR ;SETUP ERROR LOOP ADDRESS
037642 TEST30:
037642 112737 000030 001102 MOVB #30,$TSTNM ;MOVE #30 TO TEST NUMBER
037650 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
037654 012737 000031 001176 MOV #25.,$TIMES ;DO 25. ITERATIONS
  
```

;SET ATTENTION BITS FOR BOTH PORTS

```

037662 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT 64$
037670 005760 000012 66$: TST RMDS(R0) ;MAKE SURE DRIVE AVAILABLE
037674 001775 BEQ 66$
037676 012760 177777 000014 MOV #-1,RMER1(R0) ;FORCE ERRORS
037704 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
037710 013760 001226 000010 MOV PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
037716 005760 000012 64$: TST RMDS(R0) ;WAIT FOR DRIVE TO TIMEOUT
037722 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
037724 012760 177777 000014 MOV #-1,RMER1(R0) ;FORCE ERRORS ON PORT 65$
037732 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
037736 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT '64$' AGAIN
037744 005760 000012 65$: TST RMDS(R0) ;WAIT FOR DRIVE TO TIMEOUT
037750 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

037752 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
037760 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037766 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
037772 016037 000012 001126 MOV RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
040000 012737 000012 001122 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040006 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
040012 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
040020 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
040026 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
040034 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
040042 001414 BEQ 67$ ;BR IF OK
040044 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
040052 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
040060 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
040066 104010 EMT 10
040070 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
040074 000240 67$: NOP
040076 005737 001250 TST CKERR ;WAS ATTENTION SET FOR A??
040102 001402 BEQ .+6 ;YES!!
040104 000137 041276 JMP 1$ ;NO - BYPASS REST OF TEST
040110 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
040116 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040124 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
040130 016037 000012 001126 MOV RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
040136 012737 000012 001122 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040144 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
040150 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
040156 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
040164 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
040172 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
040200 001414 BEQ 69$ ;BR IF OK
  
```

```

040202 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
040210 042737 100000 001174      BIC      #ATA,$TMP4        ;CLEAR THE MASKED BITS
040216 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
040224 104010                      EMT      10
040226 005137 001250                      COM      CKERR             ;SET THE REGISTER COMPARE ERROR INDICATOR
040232 000240                      69$:    NOP
040234 005737 001250                      TST      CKERR            ;WAS ATTENTION SET FOR B??
040240 001402                      BEQ      +6               ;YES!!
040242 000137 041275                      JMP      1$               ;NO - BYPASS REST OF TEST
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

040246 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
040254 013737 001224 001242      MOV      PORTA, SEIZPT    ;STORE SEIZING PORT'S ADDRESS
040262 005060 000012                      CLR      RMDS(RO)        ;WRITE RMDS
040266 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
040274 013737 001226 001240      MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040302 013737 001226 001244      MOV      PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
040310 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
040316 010037 001122                      MOV      RO, $BDADR      ;RH/RM BASE ADDRESS
040322 062737 000012 001122      ADD      #RMDS, $BDADR   ;GENERATE BAD REGISTER ADDRESS
040330 005037 001124                      CLR      $GDDAT         ;REGISTER SHOULD BE ZERO
040334 023737 001124 001126      CMP      $GDDAT, $BDDAT  ;IS THE REGISTER ZERO
040342 001403                      BEQ      71$            ;BR IF IT IS
040344 104004                      EMT      4
040346 000137 041276                      JMP      1$             ;BYPASS REST OF THE SUBTEST
  
```

71\$:

```

040352 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
040360 013737 001224 001240      MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040366 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
040374 042737 020001 001126      BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
040402 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
040410 013737 001124 001166      MOV      $GDDAT, $TMP1   ;USE GOOD DATA AS A MASK
040416 005137 001166                      COM      $TMP1          ;COMPLEMENT THE EXPECTED STATUS
040422 013737 001126 001164      MOV      $BDDAT, $TMP0   ;SAVE THE ACTUAL STATUS
040430 043737 001166 001164      BIC      $TMP1, $TMP0    ;CLEAR UNWANTED BITS
040436 023737 001124 001164      CMP      $GDDAT, $TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
040444 001401                      BEQ      72$            ;BR IF THEY ARE
040446 104005                      EMT      5
040450 000240                      72$:    NOP
  
```

72\$:

;ISSUE NOP COMMAND TO PORT A

```

040452 012760 000001 000000      MOV      #1, RMCS1(RO)
  
```

;VERIFY THAT ATTENTION FOR PORT A CLEARED

```

040460 005037 001250                      CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040464 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
040472 012737 000012 001122      MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040500 060037 001122                      ADD      RO, $BDADR      ;ADD RH/RM BASE ADDRESS
040504 005037 001124                      CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
040510 013737 001126 001164      MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
040516 042737 077777 001164      BIC      #^CATA, $TMP0   ;SAVE SPECIFIED BITS
040524 023737 001124 001164      CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
040532 001414                      BEQ      73$            ;BR IF OK
040534 013737 001126 001174      MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
  
```



```

040542 042737 100000 001174      BIC    #ATA,$TMP4      ;CLEAR THE MASKED BITS
040550 053737 001174 001124      BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
040556 104061                      EMT    61
040560 005137 001250                      COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
040564 000240                      NOP
    
```

73\$:

;RELEASE THE DRIVE FROM PORT A

```

040566 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
040574 013737 001224 001240      MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040602 012760 000013 000000      MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

040610 005037 001254                      CLR    RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
040614 012737 000012 001122      MOV    #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
040622 060037 001122                      ADD    R0,$BDADR      ;ADD THE I/O BASE ADDRESS
040626 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
040634 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
040642 016037 000012 001170      MOV    RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
040650 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
040656 013737 001170 001164      MOV    $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
040664 042737 100100 001164      BIC    #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040672 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
040700 016037 000012 001172      MOV    RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
040706 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
040714 013737 001172 001166      MOV    $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
040722 042737 100100 001166      BIC    #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040730 023737 001164 001166      CMP    $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
040736 001006                      BNE    75$           ;BR IF NOT
040740 005737 001164                      TST    $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
040744 001045                      BNE    77$           ;BR IF NOT
040746 104046                      EMT    46
040750 000137 041150                      JMP    79$           ;BYPASS THE REST OF THE CHECKS
040754 013737 001170 001126 75$:  MOV    $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
040762 013737 001226 001240      MOV    PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040770 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
040776 005737 001164                      TST    $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
041002 001414                      BEQ    76$           ;BR IF ZERO
041004 013737 001224 001240      MOV    PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
041012 013737 001172 001126      MOV    $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
041020 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
041026 005737 001166                      TST    $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
041032 001012                      BNE    77$           ;BR IF NOT
041034 012737 177777 001254 76$:  MOV    #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
041042 012760 000011 000000      MOV    #11, RMCS1(R0) ;CLEAR THE DRIVE
041050 012760 000013 000000      MOV    #13, RMCS1(R0) ;RELEASE THE DRIVE
041056 104026                      EMT    26
041060 013737 001170 001126 77$:  MOV    $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
041066 013737 001224 001240      MOV    PORTA, PTNBR   ;CHANGE PORT NUMBER
041074 042737 100000 001126      BIC    #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
041102 023737 001124 001126      CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
041110 001401                      BEQ    78$           ;BR IF OK FROM PORT A.
041112 104007                      EMT    7
041114 013737 001172 001126 78$:  MOV    $TMP3,$BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
041122 013737 001226 001240      MOV    PORTB, PTNBR   ;CHANGE PORT NUMBER
041130 042737 100000 001126      BIC    #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
    
```

```

041136 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
041144 001401                      BEQ      79$           ;BR IF OK
041146 104007                      EMT      7
041150 000240                      NOP
79$:
;VERIFY THAT ATTENTION FOR PORT B IS STIL SET

041152 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
041160 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041166 005037 001250                      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
041172 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
041200 012737 000012 001122      MOV      #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041206 060037 001122                      ADD      RO, $BDADR ;ADD RH/RM BASE ADDRESS
041212 012737 100000 001124      MOV      #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
041220 013737 001126 001164      MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
041226 042737 077777 001164      BIC      #^CATA, $TMP0 ;SAVE SPECIFIED BITS
041234 023737 001124 001164      CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
041242 001414                      BEQ      80$           ;BR IF OK
041244 013737 001126 001174      MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
041252 042737 100000 001174      BIC      #ATA, $TMP4 ;CLEAR THE MASKED BITS
041260 053737 001174 001124      BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
041266 104062                      EMT      62
041270 005137 001250                      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
041274 000240                      NOP
041276 000004                      SCOPE
80$:
1$:
    
```

1034  
 1053  
 1054

```

*****
*TEST 31      RESET ATTENTION 'B' BY GO TEST
*
* VERIFY THAT THE 'GO BIT CLEARS ONLY THE ATTENTION BIT OF THE
* SEIZING PORT.
*
* A.  SET EACH PORT'S ATTENTION BIT, AND VERIFY THAT BOTH
* ATTENTION BITS ARE SET.
*
* B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S
* INTO RMDS.
*
* C.  ISSUE A NOP COMMAND.
*
* D.  RELEASE THE FRIVE THROUGH PORT 'B'. VERIFY THAT THE
* ATTENTION BIT FOR PORT 'B' IS RESET, AND THE
* ATTENTION BIT FOR PORT 'A' IS STIL SET.
*****
    
```

```

041300
041300 005737 001300      TST      KYBCTL ;PERFORMING ONLY SINGLE TEST ?
041304 001406                      BEQ      2$           ;BR IF NOT
041306 100002                      BPL      1$           ;BR IF JUST ENTERED TEST
041310 000137 003110      JMP      EXEC ;RETURN & GET NEXT TEST NUMBER
041314 012737 177777 001300 1$: MOV      #-1, KYBCTL ;SET SINGLE TEST INDICATOR
041322 012737 041336 001106 2$: MOV      #TEST31, $LPADR ;SETUP SCOPE LOOP ADDRESS
041330 012737 041336 001110      MOV      #TEST31, $LPERR ;SETUP ERROR LOOP ADDRESS
041336
041336 112737 000031 001102 TEST31: MOV      #31, $STNM ;MOVE #31 TO TEST NUMBER
041344 012706 001100      MOV      #STACK, SP ;LOAD THE STACK POINTER
    
```



1055  
1056

041350 012737 000031 001176 MOV #25.,\$TIMES ;;DO 25. ITERATIONS

;;SET ATTENTION BITS FOR BOTH PORTS

```

041356 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT 64$
041364 005760 000012 66$:      TST   RMDS(R0)       ;MAKE SURE DRIVE AVAILABLE
041370 001775                BEQ   66$
041372 012760 177777 000014      MOV   #-1, RMER1(R0) ;FORCE ERRORS
041400 005060 000014                CLR   RMER1(R0)       ;CLEAR THE ERRORS
041404 013760 001226 000010      MOV   PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
041412 005760 000012 64$:      TST   RMDS(R0)       ;WAIT FOR DRIVE TO TIMEOUT
041416 001775                BEQ   64$             ;BR IF DRIVE HASN'T TIMED OUT
041420 012760 177777 000014      MOV   #-1, RMER1(R0) ;FORCE ERRORS ON PORT 65$
041426 005060 000014                CLR   RMER1(R0)       ;CLEAR THE ERRORS
041432 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
041440 005760 000012 65$:      TST   RMDS(R0)       ;WAIT FOR DRIVE TO TIMEOUT
041444 001775                BEQ   65$             ;BR IF DRIVE HASN'T TIMED OUT
    
```

;;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

041446 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
041454 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041462 005037 001250                CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
041466 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
041474 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041502 060037 001122                ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
041506 012737 100000 001124      MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
041514 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
041522 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
041530 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
041536 001414                BEQ   67$           ;BR IF OK
041540 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
041546 042737 100000 001174      BIC   #ATA, $TMP4   ;CLEAR THE MASKED BITS
041554 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
041562 104010                EMT   10
041564 005137 001250                COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
041570 000240 67$:      NOP
041572 005737 001250                TST   CKERR          ;WAS ATTENTION SET FOR B??
041576 001402                BEQ   .+6           ;YES!!
041600 000137 042772                JMP   1$           ;NO - BYPASS REST OF TEST
041604 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
041612 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041620 005037 001250                CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
041624 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
041632 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041640 060037 001122                ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
041644 012737 100000 001124      MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
041652 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
041660 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
041666 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
041674 001414                BEQ   69$           ;BR IF OK
041676 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
041704 042737 100000 001174      BIC   #ATA, $TMP4   ;CLEAR THE MASKED BITS
041712 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
041720 104010                EMT   10
041722 005137 001250                COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
    
```

```

041726 000240
041730 005737 001250
041734 001402
041736 000137 042772
69$: NOP
      TST CKERR ;WAS ATTENTION SET FOR A??
      BEQ +6 ;YES!!
      JMP 1$ ;NO - BYPASS REST OF TEST
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

041742 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
041750 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
041756 005060 000012 CLR RMDS(R0) ;WRITE RMDS
041762 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
041770 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041776 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
042004 016037 000012 001126 MOV RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
042012 010037 001122 MOV RO, $BDADR ;RH/RM BASE ADDRESS
042016 062737 000012 001122 ADD #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
042024 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
042030 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
042036 001403 BEQ 71$ ;BR IF IT IS
042040 104004 EMT 4
042042 000137 042772 JMP 1$ ;BYPASS REST OF THE SUBTEST
042046
  
```

```

71$: MOVB PORTB, RMCS2(R0) ;SELECT PORT B
      MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
      MOV RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
      BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
      MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
      MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
      COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
      MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
      BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
      CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
      BEQ 72$ ;BR IF THEY ARE
      EMT 5
  
```

72\$: NOP

;ISSUE NOP COMMAND TO PORT B

```

042146 012760 000001 000000 MOV #1, RMCS1(R0)
  
```

;VERIFY THAT ATTENTION FOR PORT B CLEARED

```

042154 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042160 016037 000012 001126 MOV RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
042166 012737 000012 001122 MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042174 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
042200 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
042204 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
042212 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
042220 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
042226 001414 BEQ 73$ ;BR IF OK
042230 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
042236 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
042244 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
042252 104061 EMT 61
042254 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
042260 000240
73$: NOP
  
```







;VERIFY THAT ATTENTION FOR PORT A IS STIL SET

```

042646 113760 001224 000010      MOV  PORTA, RMCS2(RO) ;SELECT PORT A
042654 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042662 005037 001250              CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042666 016037 000012 001126      MOV  RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
042674 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042702 060037 001122              ADD  RO, $BDADR ;ADD RH/RM BASE ADDRESS
042706 012737 100000 001124      MOV  #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
042714 013737 001126 001164      MOV  $RDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
042722 042737 077777 001164      BIC  #^CATA, $TMP0 ;SAVE SPECIFIED BITS
042730 023737 001124 001164      CMP  $GDDAT, $TMP0 ;COMPARE THE BITS
042736 001414                      BEQ  80$ ;BR IF OK
042740 013737 001126 001174      MOV  $BDDAT, $TMP4 ;COPY 'BAD DATA'
042746 042737 100000 001174      BIC  #ATA, $TMP4 ;CLEAR THE MASKED BITS
042754 053737 001174 001124      BIS  $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
042762 104062                      EMT  62
042764 005137 001250              COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
042770 000240                      NOP
042772 000004                      1$: SCOPE
  
```

1057  
1071  
1072

```

*****
*TEST 32      TEST RESET ATTENTION 'A' & 'B' BY MASSBUS INIT
*
*VERIFY THAT MASSBUS CLEAR RESETS BOTH PORT'S ATTENTION BITS WHEN THE
* DRIVE IS IN NEUTRAL.
*
* A. SET THE ATTENTION BITS FOR BOTH PORTS.
*
* B. VERIFY THAT THE DRIVE IS IN NEUTRAL.
*
* C. ISSUE A MASSBUS INIT. VERIFY THAT BOTH ATTENTION BITS HAVE
* RESET.
*****
  
```

```

042774 005737 001300      TST32: TST  KYBCTL ;PERFORMING ONLY SINGLE TEST ?
042774 001406              BEQ  2$ ;BR IF NOT
043000 100002              BPL  1$ ;BR IF JUST ENTERED TEST
043004 000137 003110      JMP  EXEC ;RETURN & GET NEXT TEST NUMBER
043010 012737 177777 001300 1$: MOV  #-1, KYBCTL ;SET SINGLE TEST INDICATOR
043016 012737 043032 001106 2$: MOV  #TEST32, $LPADR ;SETUP SCOPE LOOP ADDRESS
043024 012737 043032 001110      MOV  #TEST32, $LPERR ;SETUP ERROR LOOP ADDRESS
043032              TEST32:
043032 112737 000032 001102      MOV  #32, $TSTNM ;MOVE #32 TO TEST NUMBER
043040 012706 001100              MOV  #STACK, SP ;LOAD THE STACK POINTER
043044 012737 000031 001176      MOV  #25., $TIMES ;DO 25. ITERATIONS
  
```

1073  
1108

;SET ATTENTION BITS FOR BOTH PORTS

```

043052 113760 001224 000010      66$: MOV  PORTA, RMCS2(RO) ;SELECT PORT 64$
043060 005760 000012              TST  RMDS(RO) ;MAKE SURE DRIVE AVAILABLE
043064 001775              BEQ  66$
043066 012760 177777 000014      MOV  #-1, RMER1(RO) ;FORCE ERRORS
043074 005060 000014              CLR  RMER1(RO) ;CLEAR THE ERRORS
  
```



```

043100 013760 001226 000010      MOV    PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
043106 005760 000012      64$:  1ST    RMDS(R0)      ;WAIT FOR DRIVE TO TIMEOUT
043112 001775                BEQ    64$          ;BR IF DRIVE HASN'T TIMED OUT
043114 012760 177777 000014      MOV    #-1, RMER1(R0) ;FORCE ERRORS ON PORT 65$
043122 005060 000014                CLR    RMER1(R0)     ;CLEAR THE ERRORS
043126 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
043134 005760 000012      65$:  TST    RMDS(R0)      ;WAIT FOR DRIVE TO TIMEOUT
043140 001775                BEQ    65$          ;BR IF DRIVE HASN'T TIMED OUT
    
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

043142 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
043150 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043156 005037 001250                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
043162 016037 000012 001126      MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
043170 012737 000012 001122      MOV    #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043176 060037 001122                ADD    R0, $BDADR    ;ADD RH/RM BASE ADDRESS
043202 012737 100000 001124      MOV    #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
043210 013737 001126 001164      MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
043216 042737 077777 001164      BIC    #^CATA, $TMP0 ;SAVE SPECIFIED BITS
043224 023737 001124 001164      CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
043232 001414                BEQ    67$          ;BR IF OK
043234 013737 001126 001174      MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
043242 042737 100000 001174      BIC    #ATA, $TMP4  ;CLEAR THE MASKED BITS
043250 053737 001174 001124      BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
043256 104010                EMT    10
043260 005137 001250                COM    CKERR
043264 000240      67$:  NOP
043266 005737 001250                TST    CKERR        ;WAS ATTN BIT FOR PORT A SET ?
043272 001402                BEQ    .+6         ;BR IF IT WAS
043274 000137 044250                JMP    1$          ;BYPASS REST OF TEST IF NOT
043300 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
043306 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043314 005037 001250                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
043320 016037 000012 001126      MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
043326 012737 000012 001122      MOV    #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043334 060037 001122                ADD    R0, $BDADR    ;ADD RH/RM BASE ADDRESS
043340 012737 100000 001124      MOV    #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
043346 013737 001126 001164      MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
043354 042737 077777 001164      BIC    #^CATA, $TMP0 ;SAVE SPECIFIED BITS
043362 023737 001124 001164      CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
043370 001414                BEQ    69$          ;BR IF OK
043372 013737 001126 001174      MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
043400 042737 100000 001174      BIC    #ATA, $TMP4  ;CLEAR THE MASKED BITS
043406 053737 001174 001124      BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
043414 104010                EMT    10
043416 005137 001250                COM    CKERR
043422 000240      69$:  NOP
043424 005737 001250                TST    CKERR        ;WAS ATTN BIT FOR PORT B SET ?
043430 001402                BEQ    .+6         ;BR IF IT WAS
043432 000137 044250                JMP    1$          ;BYPASS REST OF TEST IF NOT
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

043436 005037 001254                CLR    RELERR       ;CLEAR THE 'RELEASE ERROR ' INDICATOR
043442 012737 000012 001122      MOV    #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
043450 060037 001122                ADD    R0, $BDADR   ;ADD THE I/O BASE ADDRESS
    
```

```

043454 012737 111700 001124      MOV      #111700,$GDDAT      ;COMPARSION CONSTANT
043462 113760 001224 000010      MOV      PORTA, RMCS2(R0)   ;SELECT PORT A.
043470 016037 000012 001170      MOV      RMDS(R0), $TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
043476 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
043504 013737 001170 001164      MOV      $TMP2, $TMP0       ;COPY IT INTO '$TMP0'
043512 042737 100100 001164      BIC      #ATA!VV, $TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043520 113760 001226 000010      MOV      PORTB, RMCS2(R0)   ;SELECT PORT B.
043526 016037 000012 001172      MOV      RMDS(R0), $TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
043534 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
043542 013737 001172 001166      MOV      $TMP3, $TMP1       ;COPY IT INTO '$TMP1'
043550 042737 100100 001166      BIC      #ATA!VV, $TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043556 023737 001164 001166      CMP      $TMP0, $TMP1       ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
043564 001006                                BNE      71$                ;BR IF NOT
043566 005737 001164                                TST      $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
043572 001045                                BNE      73$                ;BR IF NOT
043574 104046                                EMT      46
043576 000137 043762                                JMP      75$                ;BYPASS THE REST OF THE CHECKS
043602 013737 001170 001126 71$:      MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
043610 013737 001226 001240      MOV      PORTB, PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043616 113760 001226 000010      MOV      PORTB, RMCS2(R0)   ;SELECT PORT B.
043624 005737 001164                                TST      $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
043630 001414                                BEQ      72$                ;BR IF ZERO
043632 013737 001224 001240      MOV      PORTA, PTNBR       ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043640 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
043646 113760 001224 000010      MOV      PORTA, RMCS2(R0)   ;SELECT PORT A.
043654 005737 001166                                TST      $TMP1              ;SEE IF STATUS EQ ZERO FROM PORT B.
043660 001012                                BNE      73$                ;BR IF NOT
043662 012737 177777 001254 72$:      MOV      #-1, RELERR        ;SET 'RELEASE ERROR' INDICATOR
043670 012760 000011 000000      MOV      #11, RMCS1(R0)     ;CLEAR THE DRIVE
043676 012760 000013 000000      MOV      #13, RMCS1(R0)     ;RELEASE THE DRIVE
043704 104026                                EMT      26
043706 013737 001170 001126 73$:      MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
043714 013737 001224 001240      MOV      PORTA, PTNBR       ;CHANGE PORT NUMBER
043722 023737 001124 001126      CMP      $GDDAT, $BDDAT     ;ALL BITS OK ?
043730 001401                                BEQ      74$                ;BR IF OK FROM PORT A.
043732 104007                                EMT      7
043734 013737 001172 001126 74$:      MOV      $TMP3, $BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
043742 013737 001226 001240      MOV      PORTB, PTNBR       ;CHANGE PORT NUMBER
043750 023737 001124 001126      CMP      $GDDAT, $BDDAT     ;SEE IF READ OK FROM PORT B.
043756 001401                                BEQ      75$                ;BR IF OK
043760 104007                                EMT      7
043762 000240                                NOP
043764 005737 001254 75$:      TST      RELERR             ;WAS DRIVE IN NEUTRAL ?
043770 001402                                BEQ      .+6                ;BR IF IT WAS
043772 000137 044250                                JMP      1$                 ;BYPASS RESET OF TEST

;ISSUE THE MASSBUS INIT

043776 012760 000040 000010      MOV      #CLR, RMCS2(R0)    ;ISSUE A MASSBUS INIT

;CHECK THE ATTENTION BITS OF BOTH PORTS

044004 113760 001224 000010      MOV      PORTA, RMCS2(R0)   ;SELECT PORT A
044012 013737 001224 001240      MOV      PORTA, PTNBR       ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044020 005037 001250                                CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
044024 016037 000012 001126      MOV      RMDS(R0), $BDDAT   ;GET CONTENTS OF RMDS
044032 012737 000012 001126      MOV      #RMDS, $BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
    
```



```

044040 060037 001122      ADD      RO,$BADDR      ;ADD RH/RM BASE ADDRESS
044044 005037 001124      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
044050 013737 001126 001164  MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
044056 042737 077777 001164  BIC      #^CATA,$TMP0   ;SAVE SPECIFIED BITS
044064 023737 001124 001164  CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
044072 001414      BEQ      76$           ;BR IF OK
044074 013737 001126 001174  MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
044102 042737 100000 001174  BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
044110 053737 001174 001124  BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
044116 104051      EMT      51
044120 005137 001250      COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044124 000240      NOP
044126 113760 001226 000010 76$:  MOVB    PORTB,RMCS2(RO) ;SELECT PORT B
044134 013737 001226 001240  MOV      PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044142 005037 001250      CLR      CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
044146 016037 000012 001126  MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
044154 012737 000012 001122  MOV      #RMDS,$BADDR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044162 060037 001122      ADD      RO,$BADDR      ;ADD RH/RM BASE ADDRESS
044166 005037 001124      CLR      $GDDAT        ;WHAT REGISTER SHOULD BE
044172 013737 001126 001164  MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
044200 042737 077777 001164  BIC      #^CATA,$TMP0   ;SAVE SPECIFIED BITS
044206 023737 001124 001164  CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
044214 001414      BEQ      78$           ;BR IF OK
044216 013737 001126 001174  MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
044224 042737 100000 001174  BIC      #ATA,$TMP4     ;CLEAR THE MASKED BITS
044232 053737 001174 001124  BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
044240 104051      EMT      51
044242 005137 001250      COM      CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044246 000240      NOP
044250 000004      NOP
1$:   SCOPE           ;LOOP ?
    
```

1112  
1126  
1127

```

*****
;*TEST 33      RESET ATTENTION 'A' & 'B' BY RMAS
;*
;*VERIFY THAT BOTH ATTENTION BITS CAN BE RESET BY WRITING THE
;*APPROPRIATE BIT IN THE ATTENTION SUMMARY REGISTER.
;*
;* A.  SET THE ATTENTION BITS FOR BOTH PORTS.
;*
;* B.  VERIFY THE DRIVE IS IN NEUTRAL.
;*
;* C.  WRITE THE DRIVE'S ATTENTION BIT IN RMAS. VERIFY
;*      THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMAS.
*****
    
```

```

044252 005737 001300      TST33:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
044252 001406      BEQ      2$           ;BR IF NOT
044260 100002      BPL      1$           ;BR IF JUST ENTERED TEST
044262 000137 003110      JMP      EXEC         ;RETURN & GET NEXT TEST NUMBER
044266 012737 177777 001300 1$:  MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
044274 012737 044310 001106 2$:  MOV      #TEST33,$LPADR ;SETUP SCOPE LOOP ADDRESS
044302 012737 044310 001110  MOV      #TEST33,$LPERR ;SETUP ERROR LOOP ADDRESS
044310      TEST33:
044310 112737 000033 001102  MOVB    #33,$STSTM    ;MOVE #33 TO TEST NUMBER
044316 012706 001100      MOV      #STACK,SP   ;LOAD THE STACK POINTER
    
```

1128  
1172

044322 012737 000002 001176 MOV #2.,\$TIMES ;:DO 2. ITERATIONS

;SET ATTENTION BITS FOR BOTH PORTS

```

044330 113760 001224 000010        MOVB   PORTA,RMCS2(R0) ;SELECT PORT 64$
044336 005760 000012 66$:      TST    RMDS(R0)      ;MAKE SURE DRIVE  AVAILABLE
044342 001775                BEQ    66$
044344 012760 177777 000014        MOV    #-1,RMER1(R0) ;FORCE ERRORS
044352 005060 000014                CLR    RMER1(R0)      ;CLEAR THE ERRORS
044356 013760 001226 000010        MOV    PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
044364 005760 000012 64$:      TST    RMDS(R0)      ;WAIT FOR DRIVE TO TIMEOUT
044370 001775                BEQ    64$           ;BR IF DRIVE HASN'T TIMED OUT
044372 012760 177777 000014        MOV    #-1,RMER1(R0) ;FORCE ERRORS ON PORT 65$
044400 005060 000014                CLR    RMER1(R0)      ;CLEAR THE ERRORS
044404 113760 001224 000010        MOVB   PORTA,RMCS2(R0) ;SELECT PORT '64$' AGAIN
044412 005760 000012 65$:      TST    RMDS(R0)      ;WAIT FOR DRIVE TO TIMEOUT
044416 001775                BEQ    65$           ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

044420 113760 001224 000010        MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
044426 013737 001224 001240        MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044434 005037 001250                CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
044440 016037 000012 001126        MOV    RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
044446 012737 000012 001122        MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044454 060037 001122                ADD    R0,$BDADR     ;ADD RH/RM BASE ADDRESS
044460 012737 100000 001124        MOV    #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
044466 013737 001126 001164        MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
044474 042737 077777 001164        BIC   #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044502 023737 001124 001164        CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
044510 001414                BEQ   67$           ;BR IF OK
044512 013737 001126 001174        MOV   $BDDAT,$TMP4  ;COPY 'BAD DATA'
044520 042737 100000 001174        BIC   #ATA,$TMP4   ;CLEAR THE MASKED BITS
044526 053737 001174 001124        BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
044534 104010                EMT   10
044536 005137 001250                COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044542 000240 67$:      NOP
044544 005737 001250                TST   CKERR         ;WAS ATA SET FOR A??
044550 001402                BEQ   +6           ;YES - CONTINUE
044552 000137 045374                JMP   1$           ;BYPASS REST OF TEST
044556 113760 001226 000010        MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
044564 013737 001226 001240        MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044572 005037 001250                CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
044576 016037 000012 001126        MOV    RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
044604 012737 000012 001122        MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044612 060037 001122                ADD    R0,$BDADR     ;ADD RH/RM BASE ADDRESS
044616 012737 100000 001124        MOV    #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
044624 013737 001126 001164        MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
044632 042737 077777 001164        BIC   #^CATA,$TMP0  ;SAVE SPECIFIED BITS
044640 023737 001124 001164        CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
044646 001414                BEQ   69$           ;BR IF OK
044650 013737 001126 001174        MOV   $BDDAT,$TMP4  ;COPY 'BAD DATA'
044656 042737 100000 001174        BIC   #ATA,$TMP4   ;CLEAR THE MASKED BITS
044664 053737 001174 001124        BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
044672 104010                EMT   10
044674 005137 001250                COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044700 000240 69$:      NOP
  
```



```

044702 005737 001250          TST      CKERR          ;WAS ATA SET FOR B??
044706 001402                   BEQ      +6             ;YES - CONTINUE
044710 000137 045374          JMP      i$            ;BYPASS REST OF TEST
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

044714 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
044720 012737 000012 001122   MOV      #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
044726 060037 001122          ADD      RC,$BDADR    ;ADD THE I/O BASE ADDRESS
044732 012737 111700 001124   MOV      #11700,$GDDAT ;COMPARSION CONSTANT
044740 113760 001224 000010   MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
044746 016037 000012 001170   MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044754 042737 024001 001170   BIC     #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
044762 013737 001170 001164   MOV      $TMP2, $TMP0  ;COPY IT INTO '$TMP0'
044770 042737 100100 001164   BIC     #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044776 113760 001226 000010   MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
045004 016037 000012 001172   MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
045012 042737 024001 001172   BIC     #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
045020 013737 001172 001166   MOV      $TMP3, $TMP1  ;COPY IT INTO '$TMP1'
045026 042737 100100 001166   BIC     #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045034 023737 001164 001166   CMP     $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
045042 001006                   BNE     71$           ;BR IF NOT
045044 005737 001164          TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
045050 001045                   BNE     73$           ;BR IF NOT
045052 104046                   EMT     46
045054 000137 045240          JMP     75$           ;BYPASS THE REST OF THE CHECKS
045060 013737 001170 001126 71$: MOV      $TMP2, $BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
045066 013737 001226 001240   MOV      PORTB, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045074 113760 001226 000010   MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
045102 005737 001164          TST     $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
045106 001414                   BEQ     72$           ;BR IF ZERO
045110 013737 001224 001240   MOV      PORTA, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045116 013737 001172 001126   MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
045124 113760 001224 000010   MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
045132 005737 001166          TST     $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
045136 001012                   BNE     73$           ;BR IF NOT
045140 012737 177777 001254 72$: MOV      #-1, RELERR   ;SET 'RELEASE ERROR' INDICATOR
045146 012760 000011 000000   MOV      #11, RMCS1(RO) ;CLEAR THE DRIVE
045154 012760 000013 000000   MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
045162 104026                   EMT     26
045164 013737 001170 001126 73$: MOV      $TMP2, $BDDAT  ;LOOK FOR BIT FAILURES WHEN RMDS READ
045172 013737 001224 001240   MOV      PORTA, PTNBR  ;CHANGE PORT NUMBER
045200 023737 001124 001126   CMP     $GDDAT, $BDDAT ;ALL BITS OK ?
045206 001401                   BEQ     74$           ;BR IF OK FROM PORT A.
045210 104007                   EMT     7
045212 013737 001172 001126 74$: MOV      $TMP3, $BDDAT  ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
045220 013737 001226 001240   MOV      PORTB, PTNBR  ;CHANGE PORT NUMBER
045226 023737 001124 001126   CMP     $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
045234 001401                   BEQ     75$           ;BR IF OK
045236 104007                   EMT     7
045240 000240 001254 001126 75$: NOP
045242 005737 001254          TST     RELERR        ;WAS DRIVE IN NEUTRAL??
045246 001402                   BEQ     +6             ;YES!!
045250 000137 045374          JMP     i$            ;BYPASS REST OF TEST
  
```

;WRITE THE ATTENTION BIT

```

045254 013760 001236 000016      MOV      ASR1,RMAS(RO)

;VERIFY THAT BOTH ATTENTIONS ARE RESET BY READING RMAS
045262 016037 000016 001126      MOV      RMAS(RO),%BDDAT      ;GET ATTENTION SUMMARY
045270 033737 001236 001126      BIT      ASR1,%BDDAT          ;IS THE ATTENTION RESET ??
045276 001414                      BEQ      2$                   ;YES !!
045300 010037 001122                      MOV      RO,%BDADR           ;SETUP REGISTER ADDRESS
045304 062737 000016 001122      ADD      #RMAS,%BDADR
045312 013737 001126 001124      MOV      %BDDAT,%GDDAT      ;SETUP EXPECTED DATA
045320 043737 001236 001124      BIC      ASR1,%GDDAT        ;RESET THIS DRIVES BIT
045326 104060                      EMT      60
  
```

```

045330      2$:
;WAIT FOR THE DRIVES TO RELEASE BY TIMEOUT
  
```

```

045330 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
045336 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045344 005760 000012                      3$: TST      RMDS(RO) ;MAKE SURE DEVICE IS AVAILABLE
045350 001775                      BEQ      3$
045352 113760 001226 000010      MOVB     PORTB,RMCS2(RO) ;SELECT PORT B
045360 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045366 005760 000012                      4$: TST      RMDS(RO) ;MAKE SURE DEVICE IS AVAILABLE
045372 001775                      BEQ      4$
045374 000004                      1$: SCOPE
  
```

1173  
 1186  
 1187

```

;*****
;*TEST 34      PORT 'A' ALTERNATE ATTENTION PATH TEST
;*
;*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
;*
;*  A.  SET THE ATTENTION BIT FOR PORT 'A'.
;*
;*  B.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
;*
;*  C.  READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
;*      FOR THE DRIVE IS SET.
;*****
  
```

```

045376 005737 001300      TST34: TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
045376 001406                      BEQ      2$           ;BR IF NOT
045404 100002                      BPL      1$           ;BR IF JUST ENTERED TEST
045406 000137 003110      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
045412 012737 177777 001300 1$: MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
045420 012737 045434 001106 2$: MOV      #TEST34,%LPADR ;SETUP SCOPE LOOP ADDRESS
045426 012737 045434 001110      MOV      #TEST34,%LPERR ;SETUP ERROR LOOP ADDRESS
045434
045434 112737 000034 001102 TEST34: MOVB     #34,%STSTNM ;MOVE #34 TO TEST NUMBER
045442 012706 001100      MOV      #STACK,SP    ;LOAD THE STACK POINTER
045446 012737 000031 001176      MOV      #25.,%TIMES  ;;DO 25. ITERATIONS
  
```

1188  
 1222

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

045454 113760 001224 000010      MOVB     PORTA,RMCS2(RO) ;SELECT PORT #A
045462 005060 000012      CLR      RMDS(RO)      ;SEIZE THE DRIVE
  
```



```

045466 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
045474 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
045502 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
045510 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
045514 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
045522 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
045530 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
045536 012760 177777 000014 MOV #-1,RMER1(R0) ;SET ERRORS TO FORCE ATTN BIT ON PORT A
045544 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
045550 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
045556 005760 000012 1$: TST RMDS(R0) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
045562 001775 BEQ 1$ ;BR IF STILL SEIZED BY PORT A
045564 012737 000016 001122 MOV #RMAS,$BDADR ;FORM ADDRESS OF ATTN REG IF ERROR
045572 060037 001122 ADD R0,$BDADR ;ADD THE ADDRESS BASE
045576 013737 001236 001124 MOV ASR1,$GDDAT ;GOOD DATA FOR ERROR MESSAGE
045604 013737 001236 001166 MOV ASR1,$TMP1 ;MAKE DATA COMPARE MASK
045612 005137 001166 COM $TMP1 ;COMPLEMENT IT
045616 012737 045652 001110 MOV #2$,$LPERR ;LOAD LOOP ON ERROR ADDRESS
045624 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
045632 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045640 013737 001226 001242 MOV PORTB,SEIZPT ;'SEIZED' PORT ADDRESS
045646 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT B
045652 016037 000016 001126 2$: MOV RMAS(R0),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
045660 013737 001126 001164 MOV $BDDAT,$TMP0 ;PUT CONTENTS INTO WORKING LOCATION
045666 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR OTHER BITS
045674 023737 001124 001164 CMP $GDDAT,$TMP0 ;SEE IF ATTN BIT FOR DRIVE SET
045702 001401 BEQ 3$ ;BR IF SET
045704 104053 EMT 53
045706 3$: ;RELEASE THE DRIVE FROM PORT B

045706 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
045714 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045722 012760 000013 000000 MOV #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

045730 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
045734 012737 000012 001122 MOV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
045742 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
045746 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
045754 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
045762 016037 000012 001170 MOV RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
045770 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
045776 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
046004 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046012 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
046020 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046026 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
046034 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
046042 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046050 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046056 001006 BNE 64$ ;BR IF NOT
046060 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046064 001045 BNE 66$ ;BR IF NOT
046066 104046 EMT 46
046070 000137 046270 JMP 68$ ;BYPASS THE REST OF THE CHECKS
    
```

```

046074 013737 001170 001126 64$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046102 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046110 113760 001226 000010 MOVB PORTB,RMCS2(RO) ;SELECT PORT B.
046116 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
046122 001414 BEQ 65$ ;BR IF ZERO
046124 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046132 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
046140 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A.
046146 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
046152 001012 BNE 66$ ;BR IF NOT
046154 012737 177777 001254 65$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
046162 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
046170 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
046176 104026 EMT 26
046200 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
046206 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
046214 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
046222 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
046230 001401 BEQ 67$ ;BR IF OK FROM PORT A.
046232 104007 EMT 7
046234 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
046242 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
046250 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
046256 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
046264 001401 BEQ 68$ ;BR IF OK
046266 104007 EMT 7
046270 000240 68$: NOP
046272 000004 SCOPE ;LOOP ?
    
```

1223  
1236  
1237

```

*****
*TEST 35 PORT 'B' ALTERNATE ATTENTION PATH TEST
*
*VERIFY THAT THE ALTERNATE ATTENTION REGISTER READ PATH IS OPERATIONAL.
*
* A. SET THE ATTENTION BIT FOR PORT 'B'.
*
* B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.
*****
    
```

```

046274 005737 001300 TST35: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
046274 001406 BEQ 2$ ;BR IF NOT
046300 100002 BPL 1$ ;BR IF JUST ENTERED TEST
046304 000137 003110 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
046310 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
046316 012737 046332 001106 2$: MOV #TEST35,$LPADR ;SETUP SCOPE LOOP ADDRESS
046324 012737 046332 001110 MOV #TEST35,$LPERR ;SETUP ERROR LOOP ADDRESS
046332 TEST35: MOV #35,$STNM ;MOVE #35 TO TEST NUMBER
046332 112737 000035 001102 MOV #STACK,SP ;LOAD THE STACK POINTER
046340 012706 001100 MOV #25.,$TIMES ;DO 25. ITERATIONS
046344 012737 000031 001176
    
```

1238  
1239



;CLEAR ATTENTION BITS FOR BOTH PORTS

```

046352 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
046360 005060 000012              CLR   RMDS(R0)         ;SEIZE THE DRIVE
046364 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
046372 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
046400 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
046406 005060 000012              CLR   RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
046412 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
046420 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
046426 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
046434 012760 177777 000014      MOV   #-1, RMER1(R0)  ;SET ERRORS TO FORCE ATTN BIT ON PORT B
046442 005060 000014              CLR   RMER1(R0)       ;CLEAR THE ERRORS
046446 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
046454 005760 000012      1$:  TST   RMDS(R0)       ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
046460 001775              BEQ   1$               ;BR IF STILL SEIZED BY PORT B
046462 012737 000016 001122      MOV   #RMAS, $BDADR   ;FORM ADDRESS OF ATTN REG IF ERROR
046470 060037 001122              ADD   RO, $BDADR      ;ADD THE ADDRESS BASE
046474 013737 001236 001124      MOV   ASR1, $GDDAT    ;GOOD DATA FOR ERROR MESSAGE
046502 013737 001236 001166      MOV   ASR1, $TMP1     ;MAKE DATA COMPARE MASK
046510 005137 001166              COM   $TMP1           ;COMPLEMENT IT
046514 012737 046550 001110      MOV   #2$, $LPERR     ;LOAD LOOP ON ERROR ADDRESS
046522 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
046530 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046536 013737 001224 001242      MOV   PORTA, SEIZPT   ;'SEIZED' PORT ADDRESS
046544 005060 000012              CLR   RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT A
046550 016037 000016 001126      2$:  MOV   RMAS(R0), $BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
046556 013737 001126 001164      MOV   $BDDAT, $TMP0   ;PUT CONTENTS INTO WORKING LOCATION
046564 043737 001166 001164      BIC   $TMP1, $TMP0    ;CLEAR OTHER BITS
046572 023737 001124 001164      CMP   $GDDAT, $TMP0   ;SEE IF ATTN BIT FOR DRIVE SET
046600 001401              BEQ   3$              ;BR IF SET
046602 104053              EMT   53
046604      3$:

```

;RELEASE THE DRIVE FROM PORT A

```

046604 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
046612 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046620 012760 000013 000000      MOV   #13, RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT A

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

046626 005037 001254              CLR   RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
046632 012737 000012 001122      MOV   #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
046640 060037 001122              ADD   RO, $BDADR      ;ADD THE I/O BASE ADDRESS
046644 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
046652 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
046660 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
046666 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
046674 013737 001170 001164      MOV   $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
046702 042737 100100 001164      BIC   #ATA!VV, $TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046710 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
046716 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046724 042737 024001 001172      BIC   #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
046732 013737 001172 001166      MOV   $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
046740 042737 100100 001166      BIC   #ATA!VV, $TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046746 023737 001164 001166      CMP   $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046754 001006              BNE   64$             ;BR IF NOT

```

```

046756 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046762 001045      BNE      66$        ;BR IF NOT
046764 104046      EMT      46
046766 000137 047166      JMP      68$        ;BYPASS THE REST OF THE CHECKS
046772 013737 001170 001126 64$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047000 013737 001226 001240      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047006 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B.
047014 005737 001164      TST     $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
047020 001414      BEQ     65$        ;BR IF ZERO
047022 013737 001224 001240      MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047030 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
047036 113760 001224 000010      MOV     PORTA,RMCS2(R0) ;SELECT PORT A.
047044 005737 001166      TST     $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
047050 001012      BNE     66$        ;BR IF NOT
047052 012737 177777 001254 65$:  MOV     #-1,RELERR  ;SET 'RELEASE ERROR' INDICATOR
047060 012760 000011 000000      MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
047066 012760 000013 000000      MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
047074 104026      EMT     26
047076 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
047104 013737 001224 001240      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
047112 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
047120 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
047126 001401      BEQ     67$        ;BR IF OK FROM PORT A.
047130 104007      EMT     7
047132 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
047140 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
047146 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
047154 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
047162 001401      BEQ     68$        ;BR IF OK
047164 104007      EMT     7
047166 000240 68$:  NOP
047170 000004      SCOPE      ;LOOP ?
    
```

1240  
1257  
1258

```

*****
*TEST 36      SET ATTENTION 'A' BY COMMAND TEST
*
*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
*COMMAND.
*
*  A.  ISSUE A OFFSET COMMAND THROUGH PORT 'A'.
*
*  B.  WAIT FOR THE OFFSET COMMAND TO COMPLETE ('DRY' TO BECOME
*      '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'A' IS SET AND
*      THAT THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
*
*  C.  RELEASE THE DRIVE THROUGH PORT 'A'.  VERIFY THAT THE DRIVE RETURNED
*      TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*****
    
```

```

047172
047172 005737 001300      TST36:  TST     KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
047176 001406      BEQ     2$          ;BR IF NOT
047200 100002      BPL     1$          ;BR IF JUST ENTERED TEST
047202 000137 003110      JMP     EXEC        ;RETURN & GET NEXT TEST NUMBER
047206 012737 177777 001300 1$:  MOV     #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
    
```



```

047214 012737 047230 001106 2$: MOV #TEST36,$LPADR ;SETUP SCOPE LOOP ADDRESS
047222 012737 047230 001110 MOV #TEST36,$LPERR ;SETUP ERROR LOOP ADDRESS
047230 TEST36:
047230 112737 000036 001102 MOV #36,$STNM ;MOVE #36 TO TEST NUMBER
047236 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
047242 012737 000031 001176 MOV #25.,$TIMES ;;DO 25. ITERATIONS
  
```

1259  
1288

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

047250 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
047256 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE
047262 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
047270 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
047276 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
047304 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
047310 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
047316 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
047324 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
047332 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047340 013737 001224 001242 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS
  
```

;DO A OFFSET THROUGH PORT A

```

047346 012760 000015 000000 MOV #15,RMCS1(R0) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT A
  
```

;WAIT FOR DRIVE TO FINISH

```

047354 032760 000200 000012 BIT #DRY,RMDS(R0) ;WAIT FOR DRIVE TO FINISH
047362 001774 BEQ .-6 ;BR IF NOT FINISHED
  
```

;CONFIRM THAT ATTENTION IS SET FOR PORT A

```

047364 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
047370 016037 000012 001126 MOV RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
047376 012737 000012 001122 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
047404 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
047410 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
047416 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
047424 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
047432 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
047440 001414 BEQ 64$ ;BR IF OK
047442 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
047450 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
047456 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
047464 104032 EMT 32
047466 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
047472 000240 64$: NOP
  
```

;RELEASE THE DRIVE FROM PORT A

```

047474 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
047502 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047510 012760 000013 000000 MOV #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

047516 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
047522 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
047530 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
047534 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
047542 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
047550 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
047556 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
047564 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
047572 042737 100100 001164  BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047600 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
047606 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
047614 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
047622 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
047630 042737 100100 001166  BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047636 023737 001164 001166  CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
047644 001006          BNE      66$        ;BR IF NOT
047646 005737 001164          TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
047652 001045          BNE      68$        ;BR IF NOT
047654 104046          EMT      46
047656 000137 050042          JMP      70$
047662 013737 001170 001126 66$: MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047670 013737 001226 001240  MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047676 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
047704 005737 001164          TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
047710 001414          BEQ      67$        ;BR IF ZERO
047712 013737 001224 001240  MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047720 013737 001172 001126  MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
047726 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
047734 005737 001166          TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
047740 001012          BNE      68$        ;BR IF NOT
047742 012737 177777 001254 67$: MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
047750 012760 000011 000000  MOV      #11, RMCS1(RO) ;CLEAR THE DRIVE
047756 012760 000013 000000  MOV      #13, RMCS1(RO) ;RELEASE THE DRIVE
047764 104026          EMT      26
047766 013737 001170 001126 68$: MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
047774 013737 001224 001240  MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
050002 023737 001124 001126  CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
050010 001401          BEQ      69$        ;BR IF OK FROM PORT A.
050012 104007          EMT      7
050014 013737 001172 001126 69$: MOV      $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
050022 013737 001226 001240  MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
050030 023737 001124 001126  CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
050036 001401          BEQ      70$        ;BR IF OK
050040 104007          EMT      7
050042 000240          NOP
050044 113760 001226 000010 70$: MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B
050052 013737 001226 001240  MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

```

050060 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
050064 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
050072 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
050100 060037 001122          ADD      RO,$BDADR   ;ADD RH/RM BASE ADDRESS
050104 005037 001124          CLR      $GDDAT     ;WHAT REGISTER SHOULD BE
050110 013737 001126 001164  MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
050116 042737 077777 001164  BIC      #^CATA, $TMP0 ;SAVE SPECIFIED BITS

```



```

050124 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
050132 001414                      BEQ      71$              ;BR IF OK
050134 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
050142 042737 100000 001174      BIC      #ATA,$TMP4       ;CLEAR THE MASKED BITS
050150 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
050156 104032                      EMT      32
050160 005137 001250              COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
050164 000240                      NOP
71$:
050166 000004                      SCOPE                    ;LOOP ?
    
```

1289  
1305  
1306

```

:*****
:*TEST 37      SET ATTENTION 'B' BY COMMAND TEST
:*
:*TEST THE OPERATION OF THE PORT A AND PORT B ATTENTION BITS AFTER A
:*  COMMAND.
:*
:*  A.  ISSUE A RECALIBRATE COMMAND THROUGH PORT 'B'.
:*
:*  B.  WAIT FOR THE RECALIBRATE COMMAND TO COMPLETE ('DRY' TO BECOME
:*       '1').  VERIFY THAT THE ATTENTION BIT FOR PORT 'B' IS SET AND
:*       THAT THE ATTENTION BIT FOR PORT 'A' IS NOT SET.
:*
:*  C.  RELEASE THE DRIVE THROUGH PORT 'B'.  VERIFY THAT THE DRIVE RETURNED
:*       TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
:*
:*****
    
```

```

050170
050170 005737 001300      TST      KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
050174 001406                      BEQ      2$              ;BR IF NOT
050176 100002                      BPL      1$              ;BR IF JUST ENTERED TEST
050200 000137 003110      JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
050204 012737 177777 001300  1$:      MOV      #-1,KYBCTL      ;SET SINGLE TEST INDICATOR
050212 012737 050226 001106  2$:      MOV      #TEST37,$LPADR ;SETUP SCOPE LOOP ADDRESS
050220 012737 050226 001110      MOV      #TEST37,$LPERR ;SETUP ERROR LOOP ADDRESS
050226
050226 112737 000037 001102  TEST37:  MOVB     #37,$STSTM      ;MOVE #37 TO TEST NUMBER
050234 012706 001100      MOV      #STACK,SP      ;LOAD THE STACK POINTER
050240 012737 000031 001176      MOV      #25.,$TIMES    ;DO 25. ITERATIONS
    
```

1307  
1308

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

050246 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
050254 005060 000012                      CLR      RMDS(R0)       ;SEIZE THE DRIVE
050260 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
050266 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
050274 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
050302 005060 000012                      CLR      RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
050306 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
050314 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
050322 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
050330 013737 001226 001240      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050336 013737 001226 001242      MOV      PORTB,SEIZPT   ;'SEIZED' PORT ADDRESS
    
```

;DO A OFFSET THROUGH PORT B



```

050344 012760 000015 000000      MOV      #15, RMCS1(R0)  ;ISSUE A OFFSET INSTRUCTION THROUGH PORT B
;WAIT FOR DRIVE TO FINISH

050352 032760 000200 000012      BIT      #DRY, RMDS(R0)  ;WAIT FOR DRIVE TO FINISH
050360 001774                      BEQ      #-6             ;BR IF NOT FINISHED

;CONFIRM THAT ATTENTION IS SET FOR PORT B

050362 005037 001250                CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
050366 016037 000012 001126        MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
050374 012737 000012 001122        MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
050402 060037 001122                ADD      R0, $BDADR      ;ADD RH/RM BASE ADDRESS
050406 012737 100000 001124        MOV      #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
050414 013737 001126 001164        MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
050422 042737 077777 001164        BIC      #^CATA, $TMP0   ;SAVE SPECIFIED BITS
050430 023737 001124 001164        CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
050436 001414                      BEQ      64$            ;BR IF OK
050440 013737 001126 001174        MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
050446 042737 100000 001174        BIC      #ATA, $TMP4     ;CLEAR THE MASKED BITS
050454 053737 001174 001124        BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
050462 104032                      EMT      32
050464 005137 001250                COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
050470 000240      64$: NOP

;RELEASE THE DRIVE FROM PORT B

050472 113760 001226 000010        MOV      PORTB, RMCS2(R0) ;SELECT PORT B
050500 013737 001226 001240        MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050506 012760 000013 000000        MOV      #13, RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

050514 005037 001254                CLR      RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
050520 012737 000012 001122        MOV      #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
050526 060037 001122                ADD      R0, $BDADR      ;ADD THE I/O BASE ADDRESS
050532 012737 011700 001124        MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
050540 113760 001224 000010        MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
050546 016037 000012 001170        MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
050554 042737 024001 001170        BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
050562 013737 001170 001164        MOV      $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
050570 042737 100100 001164        BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050576 113760 001226 000010        MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
050604 016037 000012 001172        MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
050612 042737 024001 001172        BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
050620 013737 001172 001166        MOV      $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
050626 042737 100100 001166        BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050634 023737 001164 001166        CMP      $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
050642 001006                      BNE      66$            ;BR IF NOT
050644 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
050650 001045                      BNE      68$            ;BR IF NOT
050652 104046                      EMT      46
050654 000137 051040                JMP      70$            ;BYPASS THE REST OF THE CHECKS
050660 013737 001170 001126      66$: MOV      $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
050666 013737 001226 001240        MOV      PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050674 113760 001226 000010        MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
  
```



```

050702 005737 001164          TST      $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
050706 001414                BEQ      67$          ;BR IF ZERO
050710 013737 001224 001240    MOV     PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050716 013737 001172 001126    MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
050724 113760 001224 000010    MOV     PORTA,RMCS2(R0) ;SELECT PORT A.
050732 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
050736 001012                BNE     68$          ;BR IF NOT
050740 012737 177777 001254 67$: MOV     #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
050746 012760 000011 000000    MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
050754 012760 000013 000000    MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
050762 104026                EMT     26
050764 013737 001170 001126 68$: MOV     $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
050772 013737 001224 001240    MOV     PORTA,PTNBR   ;CHANGE PORT NUMBER
051000 023737 001124 001126    CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
051006 001401                BEQ     69$          ;BR IF OK FROM PORT A.
051010 104007                EMT     7
051012 013737 001172 001126 69$: MOV     $TMP3,$BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
051020 013737 001226 001240    MOV     PORTB,PTNBR   ;CHANGE PORT NUMBER
051026 023737 001124 001126    CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
051034 001401                BEQ     70$          ;BR IF OK
051036 104007                EMT     7
051040 000240                NOP
051042 113760 001224 000010 70$: MOV     PORTA,RMCS2(R0) ;SELECT PORT A
051050 013737 001224 001240    MOV     PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

```

051056 005037 001250          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
051062 016037 000012 001126    MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
051070 012737 000012 001122    MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051076 060037 001122          ADD     R0,$BDADR     ;ADD RH/RM BASE ADDRESS
051102 005037 001124          CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
051106 013737 001126 001164    MOV     $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
051114 042737 077777 001164    BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
051122 023737 001124 001164    CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
051130 001414                BEQ     71$          ;BR IF OK
051132 013737 001126 001174    MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
051140 042737 100000 001174    BIC     #ATA,$TMP4    ;CLEAR THE MASKED BITS
051146 053737 001174 001124    BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
051154 104032                EMT     32
051156 005137 001250          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
051162 000240                NOP
71$:
    
```

1312  
1321  
1322  
1323

```

;*****
;*
;*VERIFY THAT A CHANGE IN UNIT READY SETS THE ATTENTION
;*   FOR BOTH PORTS.
;*
;*THIS FUNCTION IS PERFORMED DURING THE SET VOLUME VALID TEST.
;*
;*****
;*****
;*****
    
```

1324  
1325  
1334  
1335

1336

```

: *
: *VERIFY THAT ATTENTION SETS WHEN THE DRIVE SWITCHES AFTER
: *BEING RELEASED.
: *
: *THIS IS PERFORMED DURING THE "SET PORT REQUEST TEST"
: *
: *
: *****
    
```

1337  
1338  
1359  
1360

```

: *****
: *TEST 40 PORT 'A' SET VOLUME VALID TEST
: *VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
: *
: * A. WITH PORT 'A' SELECTED, RESET AND SET 'UNIT READY'
: * STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
: * IS SEIZED AND THAT 'VOLUME VALID' IS RESET AND
: * ATTENTION IS SET.
: *
: * B. ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
: * COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
: * VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
: * IS SET.
: *
: * C. RELEASE THE DRIVE FROM PORT 'A' AND SELECT THE DRIVE FOR
: * PORT 'B'. VERIFY THAT ATTENTION IS STIL SET AND THAT
: * VOLUME VALID IS STIL RESET.
: *
: * D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
: * PORT 'B' THEN RELEASE PORT 'B'.
: *****
    
```

```

051166
051166 005737 001300
051172 001406
051174 100002
051176 000137 003110
051202 012737 177777 001300
051210 012737 051224 001106
051216 012737 051224 001110
051224
051224 112737 000040 001102
051232 012706 001100
051236 012737 000031 001176
    
```

```

TST40:
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #TEST40,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST40,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST40:
MOVB #40,$STSTM ;MOVE #40 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #25,$TIMES ;;DO 25. ITERATIONS
    
```

1361  
1403

```

;SEIZE PORT A BY WRITING THE MAINTENANCE REGISTER,RMMR1. SET
;AND RESET 'MAINTENANCE UNIT READY' TO CAUSE VOLUME VALID TO
;RESET AND ATTENTION TO SET.
    
```

;SEIZE THE DRIVE THROUGH PORT A

```

051244 113760 001224 000010      MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
051252 013737 001224 001242      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
051260 012760 000001 000024      MOV    #DMD,RMMR1(R0) ;WRITE DMD INTO RMMR1
051266 013737 001226 001244      MOV    PORTB,OPPRT ;'OPPOSITE' PORT ADDRESS
051274 012760 001001 000024      MOV    #DMD!MUR,RMMR1(R0) ;SET UNIT READY
051302 012760 000000 000024      MOV    #0,RMMR1(R0) ;RESET DIAGNOSTIC MODE
    
```



```

;VERIFY THAT ATA = 1, VV = 0 FOR PORT A
051310 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051314 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
051322 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051330 060037 001122          ADD      RO, $BDADR    ;ADD RH/RM BASE ADDRESS
051334 012737 100000 001124  MOV      #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
051342 013737 001126 001164  MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051350 042737 077777 001164  BIC      #^CATA, $TMP0 ;SAVE SPECIFIED BITS
051356 023737 001124 001164  CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
051364 001414          BEQ      66$      ;BR IF OK
051366 013737 001126 001174  MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
051374 042737 100000 001174  BIC      #ATA, $TMP4   ;CLEAR THE MASKED BITS
051402 053737 001174 001124  BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051410 104064          EMT      64
051412 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051416 000240          NOP
66$:
051420 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051424 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
051432 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051440 060037 001122          ADD      RO, $BDADR    ;ADD RH/RM BASE ADDRESS
051444 005037 001124          CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
051450 013737 001126 001164  MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051456 042737 177677 001164  BIC      #^CVV, $TMP0  ;SAVE SPECIFIED BITS
051464 023737 001124 001164  CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
051472 001414          BEQ      68$      ;BR IF OK
051474 013737 001126 001174  MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
051502 042737 000100 001174  BIC      #VV, $TMP4    ;CLEAR THE MASKED BITS
051510 053737 001174 001124  BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051516 104065          EMT      65
051520 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051524 000240          NOP
68$:

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A
051526 012760 000011 000000  MOV      #11, RMCS1(RO) ;DRIVE CLEAR
051534 012760 000021 000000  MOV      #21, RMCS1(RO) ;READ IN PRESET

;VERIFY ATA = 0 AND VV = 1 FOR PORT A
051542 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051546 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
051554 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051562 060037 001122          ADD      RO, $BDADR    ;ADD RH/RM BASE ADDRESS
051566 012737 000100 001124  MOV      #VV, $GDDAT  ;WHAT REGISTER SHOULD BE
051574 013737 001126 001164  MOV      $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051602 042737 177677 001164  BIC      #^CVV, $TMP0  ;SAVE SPECIFIED BITS
051610 023737 001124 001164  CMP      $GDDAT, $TMP0 ;COMPARE THE BITS
051616 001414          BEQ      70$      ;BR IF OK
051620 013737 001126 001174  MOV      $BDDAT, $TMP4 ;COPY 'BAD DATA'
051626 042737 000100 001174  BIC      #VV, $TMP4    ;CLEAR THE MASKED BITS
051634 053737 001174 001124  BIS      $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051642 104013          EMT      13
051644 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051650 000240          NOP
70$:
051652 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051656 016037 000012 001126  MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
051664 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051672 060037 001122          ADD      RO, $BDADR    ;ADD RH/RM BASE ADDRESS
051676 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
```

```

051704 013737 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
051712 042737 024007 001164      BIC      #^C153770,$TMP0   ;SAVE SPECIFIED BITS
051720 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
051726 001414                      BEQ      72$              ;BR IF OK
051730 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
051736 042737 153770 001174      BIC      #153770,$TMP4    ;CLEAR THE MASKED BITS
051744 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
051752 104010                      EMT      10
051754 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
051760 000240                      NOP
  
```

72\$:

;RELEASE PORT A AND SELECT PORT B  
 ;VERIFY THAT ATA = 1 AND VV = 0 FOR PORT B

;RELEASE THE DRIVE FROM PORT A

```

051762 113760 001224 000010      MOVVB   PORTA,RMCS2(R0)   ;SELECT PORT A
051770 013737 001224 001240      MOV     PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051776 012760 000013 000000      MOV     #13,RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

052004 005037 001254                      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
052010 012737 000012 001122      MOV     #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
052016 060037 001122                      ADD     RO,$BDADR        ;ADD THE I/O BASE ADDRESS
052022 012737 011600 001124      MOV     #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
052030 113760 001224 000010      MOVVB   PORTA,RMCS2(R0)   ;SELECT PORT A.
052036 016037 000012 001170      MOV     RMDS(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
052044 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
052052 013737 001170 001164      MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
052060 042737 100100 001164      BIC     #ATA!VV,$TMP0     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052066 113760 001226 000010      MOVVB   PORTB,RMCS2(R0)   ;SELECT PORT B.
052074 016037 000012 001172      MOV     RMDS(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
052102 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
052110 013737 001172 001166      MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
052116 042737 100100 001166      BIC     #ATA!VV,$TMP1     ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052124 023737 001164 001166      CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052132 001006                      BNE     74$              ;BR IF NOT
052134 005737 001164                      TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052140 001037                      BNE     76$              ;BR IF NOT
052142 104046                      EMT     46
052144 000137 052330                      JMP     78$              ;BYPASS THE REST OF THE CHECKS
052150 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052156 013737 001226 001240      MOV     PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052164 113760 001226 000010      MOVVB   PORTB,RMCS2(R0)   ;SELECT PORT B.
052172 005737 001164                      TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
052176 001414                      BEQ     75$              ;BR IF ZERO
052200 013737 001224 001240      MOV     PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052206 013737 001172 001126      MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
052214 113760 001224 000010      MOVVB   PORTA,RMCS2(R0)   ;SELECT PORT A.
052222 005737 001166                      TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
052226 001004                      BNE     76$              ;BR IF NOT
052230 012737 177777 001254 75$:  MOV     #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
052236 104022                      EMT     22
052240 013737 001170 001126 76$:  MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
052246 013737 001224 001240      MOV     PORTA,PTNBR      ;CHANGE PORT NUMBER
052254 042737 100100 001126      BIC     #ATA!VV,$BDDAT   ;DON'T CHECK ATTN BIT OR VV BIT
  
```



```

052262 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;ALL BITS OK ?
052270 001401                      BEQ      77$                ;BR IF OK FROM PORT A.
052272 104007                      EMT      7
052274 013737 001172 001126 77$:  MOV      $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
052302 013737 001226 001240      MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
052310 042737 100100 001126      BIC      #ATA!VV,$BDDAT    ;DON'T CHECK ATTN BIT OR VV BIT
052316 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
052324 001401                      BEQ      78$                ;BR IF OK
052326 104007                      EMT      7
052330 000240                      NOP
052332 113760 001226 000010 78$:  MOVB     PORTB, RMCS2(RO)    ;SELECT PORT B
052340 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052346 005037 001250                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
052352 016037 000012 001126      MOV      RMDS(RO),$BDDAT   ;GET CONTENTS OF RMDS
052360 012737 000012 001122      MOV      #RMDS,$BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052366 060037 001122                      ADD      RO,$BDADR         ;ADD RH/RM BASE ADDRESS
052372 012737 100000 001124      MOV      #ATA,$GDDAT      ;WHAT REGISTER SHOULD BE
052400 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
052406 042737 077777 001164      BIC      #^CATA,$TMP0     ;SAVE SPECIFIED BITS
052414 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
052422 001414                      BEQ      79$                ;BR IF OK
052424 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
052432 042737 100000 001174      BIC      #ATA,$TMP4       ;CLEAR THE MASKED BITS
052440 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
052446 104064                      EMT      64
052450 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
052454 000240                      NOP
052456 005037 001250 79$:      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
052462 016037 000012 001126      MOV      RMDS(RO),$BDDAT   ;GET CONTENTS OF RMDS
052470 012737 000012 001122      MOV      #RMDS,$BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052476 060037 001122                      ADD      RO,$BDADR         ;ADD RH/RM BASE ADDRESS
052502 005037 001124                      CLR      $GDDAT            ;WHAT REGISTER SHOULD BE
052506 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
052514 042737 177677 001164      BIC      #^CVV,$TMP0     ;SAVE SPECIFIED BITS
052522 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
052530 001414                      BEQ      81$                ;BR IF OK
052532 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
052540 042737 000100 001174      BIC      #VV,$TMP4        ;CLEAR THE MASKED BITS
052546 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
052554 104065                      EMT      65
052556 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
052562 000240                      NOP
81$:  NOP

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B,
;THEN RELEASE PORT B
052564 012760 000011 000000      MOV      #11,RMCS1(RO)    ;DRIVE CLEAR
052572 012760 000021 000000      MOV      #21,RMCS1(RO)    ;READ IN PRESET
;RELEASE THE DRIVE FROM PORT B

052600 113760 001226 000010      MOVB     PORTB, RMCS2(RO)    ;SELECT PORT B
052606 013737 001226 001240      MOV      PORTB,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052614 012760 000013 000000      MOV      #13,RMCS1(RO)    ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

052622 005037 001254                      CLR      RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
052626 012737 000012 001122      MOV      #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT

```

```

052634 060037 001122      ADD      RO,$BDADR      ;ADD THE I/O BASE ADDRESS
052640 012737 011600 001124  MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
052646 113760 001224 000010  MOV      PORTA, RMCS2(RO) ;SELECT PORT A.
052654 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
052662 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
052670 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
052676 042737 100100 001164  BIC      #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052704 113760 001226 000010  MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
052712 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
052720 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
052726 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
052734 042737 100100 001166  BIC      #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052742 023737 001164 001166  CMP      $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052750 001006      BNE      83$ ;BR IF NOT
052752 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052756 001037      BNE      85$ ;BR IF NOT
052760 104046      EMT      46
052762 000137 053146      JMP      87$ ;BYPASS THE REST OF THE CHECKS
052766 013737 001170 001126 83$: MOV      $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052774 013737 001226 001240  MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053002 113760 001226 000010  MOV      PORTB, RMCS2(RO) ;SELECT PORT B.
053010 005737 001164      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
053014 001414      BEQ      84$ ;BR IF ZERO
053016 013737 001224 001240  MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053024 013737 001172 001126  MOV      $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
053032 113760 001224 000010  MOV      PORTA, RMCS2(RO) ;SELECT PORT A.
053040 005737 001166      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
053044 001004      BNE      85$ ;BR IF NOT
053046 012737 177777 001254 84$: MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
053054 104022      EMT      22
053056 013737 001170 001126 85$: MOV      $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
053064 013737 001224 001240  MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
053072 042737 100100 001126  BIC      #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
053100 023737 001124 001126  CMP      $GDDAT, $BDDAT ;ALL BITS OK ?
053106 001401      BEQ      86$ ;BR IF OK FROM PORT A.
053110 104007      EMT      7
053112 013737 001172 001126 86$: MOV      $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
053120 013737 001226 001240  MOV      PORTB, PTNBR ;CHANGE PORT NUMBER
053126 042737 100100 001126  BIC      #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
053134 023737 001124 001126  CMP      $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
053142 001401      BEQ      87$ ;BR IF OK
053144 104007      EMT      7
053146 000240      87$: NOP
053150 000004      87$: 50$: SCOPE
  
```

1404  
1425  
1426

```

*****
*TEST 41      PORT 'B' SET VOLUME VALID TEST
*VERIFY THAT VOLUME VALID CAN BE SET FOR THE PORT UNDER TEST.
*
* A.  WITH PORT 'B' SELECTED, RESET AND SET 'UNIT READY'
*      STATUS USING DIAGNOSTIC MODE. VERIFY THAT THE DRIVE
*      IS SEIZED AND THAT 'VOLUME VALID' IS RESET AND
*      ATTENTION IS SET.
*
* B.  ISSUE A DRIVE CLEAR COMMAND AND A READ IN PRESET
  
```



```

: * COMMAND TO THE DRIVE THAT WAS SEIZED IN STEP A.
: * VERIFY THAT ATTENTION IS RESET AND THAT VOLUME VALID
: * IS SET.
: *
: * C. RELEASE THE DRIVE FROM PORT 'B' AND SELECT THE DRIVE FOR
: * PORT 'A'. VERIFY THAT ATTENTION IS STILL SET AND THAT
: * VOLUME VALID IS STILL RESET.
: *
: * D. ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO
: * PORT 'A' THEN RELEASE PORT 'A'.
: *
: *****

```

```

053152
053152 005737 001300
053156 001406
053160 100002
053162 000137 003110
053166 012737 177777 001300
053174 012737 053210 001106
053202 012737 053210 001110
053210
053210 112737 000041 001102
053216 012706 001100
053222 012737 000031 001176

```

```

TST41:
TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
BEQ 2$ ;BR IF NOT
BPL 1$ ;BR IF JUST ENTERED TEST
JMP EXEC ;RETURN & GET NEXT TEST NUMBER
1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$: MOV #TEST41,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST41,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST41:
MOVB #41,$STNM ;MOVE #41 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #25.,$TIMES ;DO 25. ITERATIONS

```

1427  
1428

```

;SEIZE PORT B BY WRITING THE MAINTENANCE REGISTER,RMMR1. SET
;AND RESET 'MAINTENANCE UNIT READY' TO CAUSE VOLUME VALID TO
;RESET AND ATTENTION TO SET.

```

```

;SEIZE THE DRIVE THROUGH PORT B

```

```

053230 113760 001226 000010
053236 013737 001226 001242
053244 012760 000001 000024
053252 013737 001224 001244
053260 012760 001001 000024
053266 012760 000000 000024

```

```

MOVB PORTB,RMCS2(RO) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
MOV #DMD,RMMR1(RO) ;WRITE DMD INTO RMMR1
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV #DMD!MUR,RMMR1(RO) ;SET UNIT READY
MOV #0,RMMR1(RO) ;RESET DIAGNOSTIC MODE

```

```

;VERIFY THAT ATA = 1,VV = 0 FOR PORT B

```

```

053274 005037 001250
053300 016037 000012 001126
053306 012737 000012 001122
053314 060037 001122
053320 012737 100000 001124
053326 013737 001126 001164
053334 042737 077777 001164
053342 023737 001124 001164
053350 001414
053352 013737 001126 001174
053360 042737 100000 001174
053366 053737 001174 001124
053374 104064
053376 005137 001250
053402 000240
053404 005037 001250
053410 016037 000012 001126
053416 012737 000012 001122

```

```

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 66$ ;BR IF OK
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
EMT 64
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
66$: NOP
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE

```

```

053424 060037 001122      ADD    RO,$BDADR      ;ADD RH/RM BASE ADDRESS
053430 005037 001124      CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
053434 013737 001126 001164  MOV    $BDDAT,$TMPO  ;MOVE REGISTER CONTENTS TO '$TMPO'
053442 042737 177677 001164  BIC    #^CVV,$TMPO  ;SAVE SPECIFIED BITS
053450 023737 001124 001164  CMP    $GDDAT,$TMPO ;COMPARE THE BITS
053456 001414      BEQ    68$          ;BR IF OK
053460 013737 001126 001174  MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
053466 042737 000100 001174  BIC    #VV,$TMP4    ;CLEAR THE MASKED BITS
053474 053737 001174 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
053502 104065      EMT    65
053504 005137 001250      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
053510 000240      NOP

```

68\$:

```

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B
053512 012760 000011 000000  MOV    #11,RMCS1(RO) ;DRIVE CLEAR
053520 012760 000021 000000  MOV    #21,RMCS1(RO) ;READ IN PRESET

```

;VERIFY ATA = 0 AND VV = 1 FOR PORT B

```

053526 005037 001250      CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
053532 016037 000012 001126  MOV    RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
053540 012737 000012 001122  MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053546 060037 001122      ADD    RO,$BDADR    ;ADD RH/RM BASE ADDRESS
053552 012737 000100 001124  MOV    #VV,$GDDAT   ;WHAT REGISTER SHOULD BE
053560 013737 001126 001164  MOV    $BDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
053566 042737 177677 001164  BIC    #^CVV,$TMPO  ;SAVE SPECIFIED BITS
053574 023737 001124 001164  CMP    $GDDAT,$TMPO ;COMPARE THE BITS
053602 001414      BEQ    70$          ;BR IF OK
053604 013737 001126 001174  MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
053612 042737 000100 001174  BIC    #VV,$TMP4    ;CLEAR THE MASKED BITS
053620 053737 001174 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
053626 104013      EMT    13
053630 005137 001250      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
053634 000240      NOP

```

70\$:

```

053636 005037 001250      CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
053642 016037 000012 001126  MOV    RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
053650 012737 000012 001122  MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053656 060037 001122      ADD    RO,$BDADR    ;ADD RH/RM BASE ADDRESS
053662 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
053670 013737 001126 001164  MOV    $BDDAT,$TMPO ;MOVE REGISTER CONTENTS TO '$TMPO'
053676 042737 024007 001164  BIC    #^C153770,$TMPO ;SAVE SPECIFIED BITS
053704 023737 001124 001164  CMP    $GDDAT,$TMPO ;COMPARE THE BITS
053712 001414      BEQ    72$          ;BR IF OK
053714 013737 001126 001174  MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
053722 042737 153770 001174  BIC    #153770,$TMP4 ;CLEAR THE MASKED BITS
053730 053737 001174 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
053736 104010      EMT    10
053740 005137 001250      COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
053744 000240      NOP

```

72\$:

;RELEASE PORT B AND SELECT PORT A  
 ;VERIFY THAT ATA = 1 AND VV = 0 FOR PORT A

;RELEASE THE DRIVE FROM PORT B

```

053746 113760 001226 000010  MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
053754 013737 001226 001240  MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
053762 012760 000013 000000  MOV    #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

```



:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

053770 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
053774 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
054002 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
054006 012737 011600 001124  MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
054014 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
054022 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054030 042737 024001 001170  BIC     #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
054036 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
054044 042737 100100 001164  BIC     #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054052 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
054060 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054066 042737 024001 001172  BIC     #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
054074 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
054102 042737 100100 001166  BIC     #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054110 023737 001164 001166  CMP     $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054116 001006          BNE     74$          ;BR IF NOT
054120 005737 001164          TST     $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054124 001037          BNE     76$          ;BR IF NOT
054126 104046          EMT     46
054130 000137 054314          JMP     78$          ;BYPASS THE REST OF THE CHECKS
054134 013737 001170 001126 74$:  MOV     $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054142 013737 001226 001240  MOV     PORTB, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054150 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
054156 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
054162 001414          BEQ     75$          ;BR IF ZERO
054164 013737 001224 001240  MOV     PORTA, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054172 013737 001172 001126  MOV     $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
054200 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
054206 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
054212 001004          BNE     76$          ;BR IF NOT
054214 012737 177777 001254 75$:  MOV     #-1, RELERR  ;SET 'RELEASE ERROR' INDICATOR
054222 104022          EMT     22
054224 013737 001170 001126 76$:  MOV     $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
054232 013737 001224 001240  MOV     PORTA, PTNBR  ;CHANGE PORT NUMBER
054240 042737 100100 001126  BIC     #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
054246 023737 001124 001126  CMP     $GDDAT, $BDDAT ;ALL BITS OK ?
054254 001401          BEQ     77$          ;BR IF OK FROM PORT A.
054256 104007          EMT     7
054260 013737 001172 001126 77$:  MOV     $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
054266 013737 001226 001240  MOV     PORTB, PTNBR  ;CHANGE PORT NUMBER
054274 042737 100100 001126  BIC     #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
054302 023737 001124 001126  CMP     $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
054310 001401          BEQ     78$          ;BR IF OK
054312 104007          EMT     7
054314 000240          NOP
054316 113760 001224 000010 78$:  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A
054324 013737 001224 001240  MOV     PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054332 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
054336 016037 000012 001126  MOV     RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
054344 012737 000012 001122  MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054352 060037 001122          ADD     RO,$BDADR    ;ADD RH/RM BASE ADDRESS
054356 012737 100000 001124  MOV     #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
054364 013737 001126 001164  MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
054372 042737 077777 001164  BIC     #^CATA, $TMP0 ;SAVE SPECIFIED BITS
  
```

```

054400 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
054406 001414                      BEQ      79$               ;BR IF OK
054410 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
054416 042737 100000 001174      BIC      #ATA,$TMP4       ;CLEAR THE MASKED BITS
054424 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
054432 104064                      EMT      64
054434 005137 001250              COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
054440 000240                      NOP
054442 005037 001250              79$: CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
054446 016037 000012 001126      MOV      RMDS(RO),$BDDAT  ;GET CONTENTS OF RMDS
054454 012737 000012 001122      MOV      #RMDS,$BADDR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054462 060037 001122              ADD      RO,$BADDR       ;ADD RH/RM BASE ADDRESS
054466 005037 001124              CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
054472 013737 001126 001164      MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
054500 042737 177677 001164      BIC      #^CVV,$TMP0     ;SAVE SPECIFIED BITS
054506 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
054514 001414                      BEQ      81$               ;BR IF OK
054516 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
054524 042737 000100 001174      BIC      #VV,$TMP4        ;CLEAR THE MASKED BITS
054532 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
054540 104065                      EMT      65
054542 005137 001250              COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
054546 000240                      81$: NOP

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A,
;THEN RELEASE PORT A
054550 012760 000011 000000      MOV      #11,RMCS1(RO)   ;DRIVE CLEAR
054556 012760 000021 000000      MOV      #21,RMCS1(RO)   ;READ IN PRESET
;RELEASE THE DRIVE FROM PORT A

054564 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A
054572 013737 001224 001240      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054600 012760 000013 000000      MOV      #13,RMCS1(RO)   ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

054606 005037 001254              CLR      RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
054612 012737 000012 001122      MOV      #RMDS,$BADDR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
054620 060037 001122              ADD      RO,$BADDR       ;ADD THE I/O BASE ADDRESS
054624 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
054632 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
054640 016037 000012 001170      MOV      RMDS(RO),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054646 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
054654 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
054662 042737 100100 001164      BIC      #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054670 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
054676 016037 000012 001172      MOV      RMDS(RO),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054704 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
054712 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
054720 042737 100100 001166      BIC      #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054726 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054734 001006                      BNE      83$               ;BR IF NOT
054736 005737 001164              TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054742 001037                      BNE      85$               ;BR IF NOT
054744 104046                      EMT      46
054746 000137 055132              JMP      87$               ;BYPASS THE REST OF THE CHECKS
054752 013737 001170 001126      83$: MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE

```



```

054760 013737 001226 001240      MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054766 113760 001226 000010      MOVVB   PORTB,RMCS2(RO) ;SELECT PORT B.
054774 005737 001164                TST      $TMPO          ;SEE IF STATUS EQ 0 FROM PORT A.
055000 001414                BEQ      84$           ;BR IF ZERO
055002 013737 001224 001240      MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
055010 013737 001172 001126      MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
055016 113760 001224 000010      MOVVB   PORTA,RMCS2(RO) ;SELECT PORT A.
055024 005737 001166                TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
055030 001004                BNE      85$           ;BR IF NOT
055032 012737 177777 001254 84$:  MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
055040 104022                EMT      22
055042 013737 001170 001126 85$:  MOV      $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
055050 013737 001224 001240      MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
055056 042737 100100 001126      BIC      #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
055064 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
055072 001401                BEQ      86$           ;BR IF OK FROM PORT A.
055074 104007                EMT      7
055076 013737 001172 001126 86$:  MOV      $TMP3,$BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
055104 013737 001226 001240      MOV      PORTB,PTNBR    ;CHANGE PORT NUMBER
055112 042737 100100 001126      BIC      #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
055120 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
055126 001401                BEQ      87$           ;BR IF OK
055130 104007                EMT      7
055132 000240                NOP
055134 000004                50$:    SCOPE
  
```

1433  
1447  
1448

```

*****
*TEST 42      TEST PORT 'A' TIMEOUT DOES NOT RESET DRIVE
*
*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*
*  B.  WRITE 1'S INTO RMER1 THROUGH PORT 'A' TO FORCE AN ATTENTION.
*
*  C.  WAIT FOR THE DRIVE TO TIMEOUT.  VERIFY THAT THE DRIVE RETURNED TO
*      NEUTRAL; THAT ATTENTION IS SET FOR PORT 'A' AND NOT SET FOR
*      PORT 'B'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.
*****
  
```

```

055136 005737 001300      TST42:  TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
055136 001406                BEQ      2$           ;BR IF NOT
055142 100002                BPL      1$           ;BR IF JUST ENTERED TEST
055146 000137 003110      JMP      EXEC         ;RETURN & GET NEXT TEST NUMBER
055152 012737 177777 001300 1$:    MOV      #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
055160 012737 055174 001106 2$:    MOV      #TEST42,$LPADR ;SETUP SCOPE LOOP ADDRESS
055166 012737 055174 001110      MOV      #TEST42,$LPERR ;SETUP ERROR LOOP ADDRESS
055174                TEST42:  MOVVB   #42,$STNM     ;MOVE #42 TO TEST NUMBER
055174 112737 000042 001102      MOV      #STACK,SP    ;LOAD THE STACK POINTER
055202 012706 001100                MOV      #2.,$TIMES   ;;DO 2. ITERATIONS
055206 012737 000002 001176
  
```

1449  
1500

;CLEAR ATTENTION BITS FOR BOTH PORTS



```

055214 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
055222 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE
055226 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
055234 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
055242 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
055250 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
055254 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
055262 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

055270 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
055276 013737 001224 001242      MOV   PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
055304 005060 000012              CLR   RMDS(R0)        ;WRITE RMDS
055310 013737 001226 001244      MOV   PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS

;FORCE AN ATTENTION BY SETTING ERRORS.

055316 012760 177777 000014      MOV   #-1, RMER1(R0)  ;SET ERROR BITS

;START THE TIMER

055324 005037 001256              CLR   TIME            ;CLEAR THE ELAPSED TIME COUNTER
055330 012737 003720 001260      MOV   #2000., WATCH   ;SET WATCH TO 2000. MS
055336 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
055344 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;WAIT FOR DRIVE TO TIMEOUT

055352 005760 000012      1$:   TST   RMDS(R0)    ;WAIT FOR THE DRIVE TO BE RELEASED
055356 001004              BNE   2$             ;BR IF DRIVE RELEASED
055360 005737 001260      TST   WATCH          ;WATCH AT ZERO ?
055364 001372              BNE   1$             ;BR IF NOT
055366 104036              EMT   36
055370
055370 113760 001224 000010      2$:   MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
055376 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;THE ERROR BIT ('ERR') IN RMDS SHOULD STILL BE SET

055404 005037 001250              CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
055410 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
055416 012737 000012 001122      MOV   #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055424 060037 001122              ADD   RO, $BDADR      ;ADD RH/RM BASE ADDRESS
055430 012737 040000 001124      MOV   #ERR, $GDDAT    ;WHAT REGISTER SHOULD BE
055436 013737 001126 001164      MOV   $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
055444 042737 137777 001164      BIC   #^CERR, $TMP0   ;SAVE SPECIFIED BITS
055452 023737 001124 001164      CMP   $GDDAT, $TMP0   ;COMPARE THE BITS
055460 001414              BEQ   66$            ;BR IF OK
055462 013737 001126 001174      MOV   $BDDAT, $TMP4   ;COPY 'BAD DATA'
055470 042737 040000 001174      BIC   #ERR, $TMP4     ;CLEAR THE MASKED BITS
055476 053737 001174 001124      BIS   $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
055504 104023              EMT   23
055506 005137 001250              COM   CKERR
055512 000240      66$:  NOP
  
```



;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

055514 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
055520 016037 000014 001126  MOV      RMER1(RO),%BDDAT ;GET CONTENTS OF RMER1
055526 012737 000014 001122  MOV      #RMER1,%BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055534 060037 001122          ADD      RO,%BDDADR     ;ADD RH/RM BASE ADDRESS
055540 012737 177777 001124  MOV      #177777,%GDDAT ;WHAT REGISTER SHOULD BE
055546 023737 001124 001126  CMP      %GDDAT,%BDDAT  ;IS THE REGISTER OK ?
055554 001403          BEQ      68$           ;BR IF OK
055556 104010          EMT      10
055560 005137 001250          COM      CKERR
055564 000240          NOP
    
```

;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

```

055566 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
055572 016037 000012 001126  MOV      RMDS(RO),%BDDAT ;GET CONTENTS OF RMDS
055600 012737 000012 001122  MOV      #RMDS,%BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055606 060037 001122          ADD      RO,%BDDADR     ;ADD RH/RM BASE ADDRESS
055612 012737 100000 001124  MOV      #ATA,%GDDAT    ;WHAT REGISTER SHOULD BE
055620 013737 001126 001164  MOV      %BDDAT,%STMP0  ;MOVE REGISTER CONTENTS TO 'STMP0'
055626 042737 077777 001164  BIC      #^CATA,%STMP0  ;SAVE SPECIFIED BITS
055634 023737 001124 001164  CMP      %GDDAT,%STMP0  ;COMPARE THE BITS
055642 001414          BEQ      70$           ;BR IF OK
055644 013737 001126 001174  MOV      %BDDAT,%STMP4  ;COPY 'BAD DATA'
055652 042737 100000 001174  BIC      #ATA,%STMP4    ;CLEAR THE MASKED BITS
055660 053737 001174 001124  BIS      %STMP4,%GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
055666 104041          EMT      41
055670 005137 001250          COM      CKERR
055674 000240          NOP
    
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

055676 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
055702 012737 000012 001122  MOV      #RMDS,%BDDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
055710 060037 001122          ADD      RO,%BDDADR     ;ADD THE I/O BASE ADDRESS
055714 012737 051700 001124  MOV      #51700,%GDDAT  ;COMPARSION CONSTANT
055722 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
055730 016037 000012 001170  MOV      RMDS(RO),%STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
055736 042737 024001 001170  BIC      #PIP!WRL!OM,%STMP2 ;CLEAR DONT CARES
055744 013737 001170 001164  MOV      %STMP2,%STMP0  ;COPY IT INTO 'STMP0'
055752 042737 100100 001164  BIC      #ATA!VV,%STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055760 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
055766 016037 000012 001172  MOV      RMDS(RO),%STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
055774 042737 024001 001172  BIC      #PIP!WRL!OM,%STMP3 ;CLEAR DONT CARES
056002 013737 001172 001166  MOV      %STMP3,%STMP1  ;COPY IT INTO 'STMP1'
056010 042737 100100 001166  BIC      #ATA!VV,%STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
056016 023737 001164 001166  CMP      %STMP0,%STMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
056024 001006          BNE      72$           ;BR IF NOT
056026 005737 001164          TST      %STMP0
056032 001045          BNE      74$           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
056034 104046          EMT      46
056036 000137 056236          JMP      76$
056042 013737 001170 001126 72$: MOV      %STMP2,%BDDAT  ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
056050 013737 001226 001240  MOV      PORTB,%PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056056 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
056064 005737 001164          TST      %STMP0
    
```

```

056070 001414          BEQ      73$          ;BR IF ZERO
056072 013737 001224 001240  MOV     PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056100 013737 001172 001126  MOV     $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
056106 113760 001224 000010  MOVNB  PORTA,RMCS2(RO) ;SELECT PORT A.
056114 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
056120 001012          BNE     74$          ;BR IF NOT
056122 012737 177777 001254 73$:  MOV     #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
056130 012760 000011 000000  MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
056136 012760 000013 000000  MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
056144 104026          EMT     26
056146 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
056154 013737 001224 001240  MOV     PORTA,PTNBR    ;CHANGE PORT NUMBER
056162 042737 100000 001126  BIC     #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
056170 023737 001124 001126  CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
056176 001401          BEQ     75$          ;BR IF OK FROM PORT A.
056200 104007          EMT     7
056202 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
056210 013737 001226 001240  MOV     PORTB,PTNBR    ;CHANGE PORT NUMBER
056216 042737 100000 001126  BIC     #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
056224 023737 001124 001126  CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
056232 001401          BEQ     76$          ;BR IF OK
056234 104007          EMT     7
056236 000240          76$:  NOP
  
```

;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

```

056240 113760 001226 000010  MOVNB  PORTB,RMCS2(RO) ;SELECT PORT B
056246 013737 001226 001240  MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
056254 005037 001250          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
056260 016037 000012 001126  MOV     RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
056266 012737 000012 001126  MOV     #RMDS,$BADDR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056274 060037 001122          ADD     RO,$BADDR      ;ADD RH/RM BASE ADDRESS
056300 005037 001124          CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
056304 013737 001126 001164  MOV     $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
056312 042737 077777 001164  BIC     #^CATA,$TMP0   ;SAVE SPECIFIED BITS
056320 023737 001124 001164  CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
056326 001414          BEQ     77$          ;BR IF OK
056330 013737 001126 001174  MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
056336 042737 100000 001174  BIC     #ATA,$TMP4     ;CLEAR THE MASKED BITS
056344 053737 001174 001124  BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
056352 104052          EMT     52
056354 005137 001250          COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
056360 000240          77$:  NOP
  
```

;CLEAR ATTENTION BIT FOR PORT A

```

056362 113760 001224 000010  MOVNB  PORTA,RMCS2(RO) ;SELECT PORT #A
056370 005060 000012          CLR     RMDS(RO)      ;SEIZE THE DRIVE
056374 012760 000011 000000  MOV     #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
056402 012760 000013 000000  MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
056410 000004          3$:  SCOPE          ;LOOP ?
  
```

1501  
1515  
1516

```

*****
;*TEST 43      TEST PORT 'B' TIMEOUT DOES NOT RESET DRIVE
;*
;*VERIFY THAT PORT TIMEOUT DOES NOT INITIALIZE THE DRIVE.
  
```



- \* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- \* B. WRITE 1'S INTO RMER1 THROUGH PORT 'B'.
- \* C. WAIT FOR THE DRIVE TO TIMEOUT. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL; THAT ATTENTION IS SET FOR PORT 'B' AND IS NOT SET FOR PORT 'A'; AND THAT BOTH PORTS SEE 1'S IN THE ERROR REGISTER.

\*\*\*\*\*

056412  
 056412 005737 001300  
 056416 001406  
 056420 100002  
 056422 000137 003110  
 056426 012737 177777 001300  
 056434 012737 056450 001106  
 056442 012737 056450 001110  
 056450  
 056450 112737 000043 001102  
 056456 012706 001100  
 056462 012737 000002 001176  
 1517  
 1518

```
TST43:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOV      #TEST43,$LPADR ;SETUP SCOPE LOOP ADDRESS
          MOV      #TEST43,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST43:
MOVB     #43,$TSTNM  ;MOVE #43 TO TEST NUMBER
MOV      #STACK,SP  ;LOAD THE STACK POINTER
MOV      #2.,$TIMES ;DO 2. ITERATIONS
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```
056470 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
056476 005060 000012              CLR      RMDS(R0)       ;SEIZE THE DRIVE
056502 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
056510 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
056516 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
056524 005060 000012              CLR      RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
056530 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
056536 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
```

;SEIZE THE DRIVE THROUGH PORT B

```
056544 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
056552 013737 001226 001242      MOV      PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
056560 005060 000012              CLR      RMDS(R0)       ;WRITE RMDS
056564 013737 001224 001244      MOV      PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
```

;FORCE AN ATTENTION BY SETTING ERRORS.

```
056572 012760 177777 000014      MOV      #-1,RMER1(R0)  ;SET ERROR BITS
```

;START THE TIMER

```
056600 005037 001256              CLR      TIME           ;CLEAR THE ELAPSED TIME COUNTER
056604 012737 003720 001260      MOV      #2000.,WATCH  ;SET WATCH TO 2000. MS
056612 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
056620 013737 001224 001240      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

;WAIT FOR DRIVE TO TIMEOUT

```
056626 005760 000012      1$:      TST      RMDS(R0)      ;WAIT FOR THE DRIVE TO BE RELEASED
056632 001004              BNE     2$              ;BR IF DRIVE RELEASED
```

```

056634 005737 001260          TST    WATCH          ;WATCH AT ZERO ?
056640 001372                   BNE    1$              ;BR IF NOT
056642 104036                   EMT    36
056644                                     2$:
056644 113760 001226 000010    MOVB   PORTB, RMCS2(RO) ;SELECT PORT B
056652 013737 001226 001240    MOV    PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

;THE ERROR BIT ('ERR') IN RMDS SHOULD STILL BE SET

```

056660 005037 001250          CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
056664 016037 000012 001126    MOV    RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
056672 012737 000012 001122    MOV    #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056700 060037 001122          ADD    RO, $BDADR      ;ADD RH/RM BASE ADDRESS
056704 012737 040000 001124    MOV    #ERR, $GDDAT    ;WHAT REGISTER SHOULD BE
056712 013737 001126 001164    MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
056720 042737 137777 001164    BIC    #^CERR, $TMP0   ;SAVE SPECIFIED BITS
056726 023737 001124 001164    CMP    $GDDAT, $TMP0   ;COMPARE THE BITS
056734 001414                   BEQ    66$             ;BR IF OK
056736 013737 001126 001174    MOV    $BDDAT, $TMP4   ;COPY 'BAD DATA'
056744 042737 040000 001174    BIC    #ERR, $TMP4     ;CLEAR THE MASKED BITS
056752 053737 001174 001124    BIS    $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
056760 104023                   EMT    23
056762 005137 001250          COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
056766 000240                                     66$:
  
```

;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

056770 005037 001250          CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
056774 016037 000014 001126    MOV    RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
057002 012737 000014 001122    MOV    #RMER1, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057010 060037 001122          ADD    RO, $BDADR      ;ADD RH/RM BASE ADDRESS
057014 012737 177777 001124    MOV    #177777, $GDDAT  ;WHAT REGISTER SHOULD BE
057022 023737 001124 001126    CMP    $GDDAT, $BDDAT  ;IS THE REGISTER OK ?
057030 001403                   BEQ    68$             ;BR IF OK
057032 104010                   EMT    10
057034 005137 001250          COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
057040 000240                                     68$:
  
```

;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

```

057042 005037 001250          CLR    CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
057046 016037 000012 001126    MOV    RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
057054 012737 000012 001122    MOV    #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057062 060037 001122          ADD    RO, $BDADR      ;ADD RH/RM BASE ADDRESS
057066 012737 100000 001124    MOV    #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
057074 013737 001126 001164    MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
057102 042737 077777 001164    BIC    #^CATA, $TMP0   ;SAVE SPECIFIED BITS
057110 023737 001124 001164    CMP    $GDDAT, $TMP0   ;COMPARE THE BITS
057116 001414                   BEQ    70$             ;BR IF OK
057120 013737 001126 001174    MOV    $BDDAT, $TMP4   ;COPY 'BAD DATA'
057126 042737 100000 001174    BIC    #ATA, $TMP4     ;CLEAR THE MASKED BITS
057134 053737 001174 001124    BIS    $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
057142 104041                   EMT    41
057144 005137 001250          COM    CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
057150 000240                                     70$:
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL



```

057152 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
057156 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
057164 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
057170 012737 051700 001124  MOV      #51700,$GDDAT ;COMPARISON CONSTANT
057176 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
057204 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
057212 042737 024001 001170  BIC     #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
057220 013737 001170 001164  MOV      $TMP2, $TMP0   ;COPY IT INTO '$TMP0'
057226 042737 100100 001164  BIC     #ATA!VV, $TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
057234 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
057242 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
057250 042737 024001 001172  BIC     #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
057256 013737 001172 001166  MOV      $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
057264 042737 100100 001166  BIC     #ATA!VV, $TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
057272 023737 001164 001166  CMP     $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
057300 001006          BNE     72$           ;BR IF NOT
057302 005737 001164          TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
057306 001045          BNE     74$           ;BR IF NOT
057310 104046          EMT     46
057312 000137 057512          JMP     76$           ;BYPASS THE REST OF THE CHECKS
057316 013737 001170 001126 72$:  MOV     $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
057324 013737 001226 001240  MOV     PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057332 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B.
057340 005737 001164          TST     $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
057344 001414          BEQ     73$           ;BR IF ZERO
057346 013737 001224 001240  MOV     PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057354 013737 001172 001126  MOV     $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
057362 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
057370 005737 001166          TST     $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
057374 001012          BNE     74$           ;BR IF NOT
057376 012737 177777 001254 73$:  MOV     #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
057404 012760 000011 000000  MOV     #11, RMCS1(RO) ;CLEAR THE DRIVE
057412 012760 000013 000000  MOV     #13, RMCS1(RO) ;RELEASE THE DRIVE
057420 104026          EMT     26
057422 013737 001170 001126 74$:  MOV     $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
057430 013737 001224 001240  MOV     PORTA, PTNBR   ;CHANGE PORT NUMBER
057436 042737 100000 001126  BIC     #ATA, $BDDAT   ;DON'T CHECK THE ATTN BIT
057444 023737 001124 001126  CMP     $GDDAT, $BDDAT ;ALL BITS OK ?
057452 001401          BEQ     75$           ;BR IF OK FROM PORT A.
057454 104007          EMT     7
057456 013737 001172 001126 75$:  MOV     $TMP3, $BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
057464 013737 001226 001240  MOV     PORTB, PTNBR   ;CHANGE PORT NUMBER
057472 042737 100000 001126  BIC     #ATA, $BDDAT   ;DON'T CHECK THE ATTN BIT
057500 023737 001124 001126  CMP     $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
057506 001401          BEQ     76$           ;BR IF OK
057510 104007          EMT     7
057512 000240          76$:  NOP

```

;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

```

057514 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A
057522 013737 001224 001240  MOV     PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
057530 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
057534 016037 000012 001126  MOV     RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
057542 012737 000012 001122  MOV     #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057550 060037 001122          ADD     RO, $BDADR    ;ADD RH/RM BASE ADDRESS

```

```

057554 005037 001124          CLR    $GDDAT          ;WHAT REGISTER SHOULD BE
057560 013737 001126 001164    MOV    $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
057566 042737 077777 001164    BIC    #^CATA,$TMP0  ;SAVE SPECIFIED BITS
057574 023737 001124 001164    CMP    $GDDAT,$TMP0  ;COMPARE THE BITS
057602 001414                    BEQ    77$            ;BR IF OK
057604 013737 001126 001174    MOV    $BDDAT,$TMP4   ;COPY 'BAD DATA'
057612 042737 100000 001174    BIC    #ATA,$TMP4    ;CLEAR THE MASKED BITS
057620 053737 001174 001124    BIS    $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
057626 104052                    EMT    52
057630 005137 001250          COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
057634 000240          77$:  NOP
    
```

;CLEAR ATTENTION BIT FOR PORT B

```

057636 113760 001226 000010    MOVB   PORTB,RMCS2(R0) ;SELECT PORT #B
057644 005060 000012                    CLR    RMDS(R0)        ;SEIZE THE DRIVE
057650 012760 000011 000000    MOV    #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
057656 012760 000013 000000    MOV    #13,RMCS1(R0)  ;RELEASE THE DRIVE
057664 000004          3$:  SCOPE          ;LOOP ?
    
```

1519  
1535  
1536

```

:*****
:*TEST 44          PORT 'A' RETRIGGER BY DEMAND TEST
:*
:*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
:*
:*  A.  SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
:*
:*  B.  WAIT 500 MS AND READ RMDS THROUGH PORT 'A'.
:*
:*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
:*      TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
:*
:*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
:*      BIT IS SET.
:*****
    
```

```

057666 005737 001300          TST44:  TST    KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
057666 005737 001300          BEQ    2$              ;BR IF NOT
057672 001406                    BPL    1$              ;BR IF JUST ENTERED TEST
057674 100002                    JMP    EXEC            ;RETURN & GET NEXT TEST NUMBER
057676 000137 003110          1$:  MOV    #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
057702 012737 177777 001300    2$:  MOV    #TEST44,$LPADR ;SETUP SCOPE LOOP ADDRESS
057710 012737 057724 001106    MOV    #TEST44,$LPERR  ;SETUP ERROR LOOP ADDRESS
057716 012737 057724 001110          TEST44:  MOVB   #44,$TSTNM      ;MOVE #44 TO TEST NUMBER
057724 112737 000044 001102    MOV    #STACK,SP      ;LOAD THE STACK POINTER
057732 012706 001100          MOV    #2.,$TIMES     ;DO 2. ITERATIONS
057736 012737 000002 001176
    
```

1537  
1572

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

057744 113760 001224 000010    MOVB   PORTA,RMCS2(R0) ;SELECT PORT #A
057752 005060 000012                    CLR    RMDS(R0)        ;SEIZE THE DRIVE
057756 012760 000011 000000    MOV    #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
057764 012760 000013 000000    MOV    #13,RMCS1(R0)  ;RELEASE THE DRIVE
    
```



```

057772 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
060000 005060 000012              CLR   RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
060004 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
060012 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

060020 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
060026 013737 001224 001242      MOV   PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
060034 005060 000012              CLR   RMDS(R0)         ;WRITE RMDS
060040 013737 001226 001244      MOV   PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS

;START THE TIMER

060046 005037 001256              CLR   TIME             ;CLEAR THE ELAPSED TIME COUNTER
060052 012737 000764 001260      MOV   #500., WATCH    ;SET WATCH TO TIM. MS
060060 005737 001260 1$:          TST   WATCH            ;WATCH EQUAL TO ZERO
060064 001375 1$                   BNE   1$               ;BR IF NOT

;START THE TIMER

060066 005037 001256              CLR   TIME             ;CLEAR THE ELAPSED TIME COUNTER
060072 012737 003720 001260      MOV   #2000., WATCH   ;SET WATCH TO 2000. MS

;RETRIGGER THE TIMEOUT ONE-SHOT

060100 005760 000012              TST   RMDS(R0)        ;RETRIGGER THE ONE-SHOT
060104 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
060112 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060120 005760 000012 2$:          TST   RMDS(R0)        ;WAIT FOR TIMEOUT
060124 001004 3$                   BNE   3$               ;BR IF TIMEOUT OCCURRED
060126 005737 001260              TST   WATCH            ;WATCH EQUAL TO ZERO ?
060132 001372 2$                   BNE   2$               ;BR IF NOT
060134 104036 36                    EMT   36
060136 013737 001256 001276 3$:  MOV   TIME, TIMES     ;SAVE THE ELAPSED TIME VALUE

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060144 005037 001254              CLR   RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
060150 012737 000012 001122      MOV   #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
060156 060037 001122              ADD   R0, $BDADR      ;ADD THE I/O BASE ADDRESS
060162 012737 011700 001124      MOV   #MQL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
060170 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
060176 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
060204 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
060212 013737 001170 001164      MOV   $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
060220 042737 100100 001164      BIC   #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060226 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
060234 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
060242 042737 024001 001172      BIC   #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
060250 013737 001172 001166      MOV   $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
060256 042737 100100 001166      BIC   #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060264 023737 001164 001166      CMP   $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
060272 001006 66$                   BNE   66$              ;BR IF NOT
060274 005737 001164              TST   $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
060300 001045 68$                   BNE   68$              ;BR IF NOT
060302 104046 46                    EMT   46
  
```

```

060304 000137 060470          JMP      70$          ;BYPASS THE REST OF THE CHECKS
060310 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
060316 013737 001226 001240      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060324 113760 001226 000010      MOVVB  PORTB,RMCS2(R0) ;SELECT PORT B.
060332 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
060336 001414          BEQ     67$          ;BR IF ZERO
060340 013737 001224 001240      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060346 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
060354 113760 001224 000010      MOVVB  PORTA,RMCS2(R0) ;SELECT PORT A.
060362 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
060366 001012          BNE     68$          ;BR IF NOT
060370 012737 177777 001254 67$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
060376 012760 000011 000000      MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
060404 012760 000013 000000      MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
060412 104022          EMT     22
060414 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
060422 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
060430 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
060436 001401          BEQ     69$          ;BR IF OK FROM PORT A.
060440 104007          EMT     7
060442 013737 001172 001126 69$:  MOV     $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
060450 013737 001226 001240      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
060456 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
060464 001401          BEQ     70$          ;BR IF OK
060466 104007          EMT     7
060470 000240          70$:  NOP
    
```

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

060472 023737 001276 001264      CMP     TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
060500 003004          BGT     4$           ;BR IF GREATER
060502 023737 001276 001266      CMP     TIMES,TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
060510 002001          BGE     .+4         ;BR IF NOT
060512          4$:  EMT     25
060512 104025          SCOPE
060514 000004          ;LOOP ?
    
```

1573  
1589  
1590

```

*****
*TEST 45      PORT 'B' RETRIGGER BY DEMAND TEST
*
*VERIFY THAT THE PORT TIMEOUT ONE-SHOT CAN BE RETRIGGERED BY MASSBUS DEMAND.
*
*  A.  SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
*
*  B.  WAIT 500 MS AND WRITE 0'B INTO RMDS THROUGH PORT 'A'.
*
*  C.  VERIFY THAT THE TIMEOUT OCCURS WITHIN + OR - 25% OF THE SPECIFIED
*      TIME. (THE MEASUREMENT IS MADE FROM STEP 'B'.)
*
*  D.  VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION
*      BIT IS SET.
*****
    
```

```

060516 005737 001300      TST45:  TST     KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
060516 005737 001300      BEQ     2$           ;BR IF NOT
060522 001406
    
```



1591  
1592

```

060524 100002          BPL      1$          ;BR IF JUST ENTERED TEST
060526 000137 003110  JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
060532 012737 177777 001300 1$:      MOV      #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
060540 012737 060554 001106 2$:      MOV      #TEST45,$LPADR ;SETUP SCOPE LOOP ADDRESS
060546 012737 060554 001110      MOV      #TEST45, PERR ;SETUP ERROR LOOP ADDRESS
060554          TEST45:
060554 112737 000045 001102      MOVB     #45,$TSTNM     ;MOVE #45 TO TEST NUMBER
060562 012706 001100      MOV      #STACK,SP     ;LOAD THE STACK POINTER
060566 012737 000002 001176      MOV      #2.,$TIMES    ;;DO 2. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

060574 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
060602 005060 000012          CLR      RMDS(R0)       ;SEIZE THE DRIVE
060606 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
060614 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE
060622 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
060630 005060 000012          CLR      RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
060634 012760 000011 000000      MOV      #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
060642 012760 000013 000000      MOV      #13,RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

060650 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
060656 013737 001226 001242      MOV      PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
060664 005060 000012          CLR      RMDS(R0)       ;WRITE RMDS
060670 013737 001224 001244      MOV      PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS

;START THE TIMER

060676 005037 001256          CLR      TIME           ;CLEAR THE ELAPSED TIME COUNTER
060702 012737 000764 001260      MOV      #500.,WATCH    ;SET WATCH TO TIM. MS
060710 005737 001260 1$:      TST      WATCH         ;WATCH EQUAL TO ZERO
060714 001375          BNE     1$             ;BR IF NOT

;START THE TIMER

060716 005037 001256          CLR      TIME           ;CLEAR THE ELAPSED TIME COUNTER
060722 012737 003720 001260      MOV      #2000.,WATCH   ;SET WATCH TO 2000. MS

;RETRIGGER THE TIMEOUT ONE-SHOT

060730 005760 000012          TST      RMDS(R0)       ;RETRIGGER THE ONE-SHOT
060734 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
060742 013737 001224 001240      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060750 005760 000012 2$:      TST      RMDS(R0)       ;WAIT FOR TIMEOUT
060754 001004          BNE     3$             ;BR IF TIMEOUT OCCURRED
060756 005737 001260      TST      WATCH         ;WATCH EQUAL TO ZERO ?
060762 001372          BNE     2$             ;BR IF NOT
060764 104036          EMT      36
060766 013737 001256 001276 3$:      MOV      TIME,TIMES    ;SAVE THE ELAPSED TIME VALUE

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060774 005037 001254          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
061000 012737 000012 001122      MOV      #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
  
```

```

061006 060037 001122      ADD      RO,$BDADR      ;ADD THE I/O BASE ADDRESS
061012 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
061020 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
061026 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
061034 042737 024001 001170  BIC     #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
061042 013737 001170 001164  MOV     $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
061050 042737 100100 001164  BIC     #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061056 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
061064 016037 000012 001172  MOV     RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061072 042737 024001 001172  BIC     #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
061100 013737 001172 001166  MOV     $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
061106 042737 100100 001166  BIC     #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061114 023737 001164 001166  CMP     $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
061122 001006      BNE     66$ ;BR IF NOT
061124 005737 001164      TST     $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
061130 001045      BNE     68$ ;BR IF NOT
061132 104046      EMT     46
061134 000137 061320      JMP     70$ ;BYPASS THE REST OF THE CHECKS
061140 013737 001170 001126 66$: MOV     $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
061146 013737 001226 001240  MOV     PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061154 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
061162 005737 001164      TST     $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
061166 001414      BEQ     67$ ;BR IF ZERO
061170 013737 001224 001240  MOV     PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061176 013737 001172 001126  MOV     $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
061204 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
061212 005737 001166      TST     $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
061216 001012      BNE     68$ ;BR IF NOT
061220 012737 177777 001254 67$: MOV     #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
061226 012760 000011 000000  MOV     #11, RMCS1(RO) ;CLEAR THE DRIVE
061234 012760 000013 000000  MOV     #13, RMCS1(RO) ;RELEASE THE DRIVE
061242 104022      EMT     22
061244 013737 001170 001126 68$: MOV     $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
061252 013737 001224 001240  MOV     PORTA, PTNBR ;CHANGE PORT NUMBER
061260 023737 001124 001126  CMP     $GDDAT, $BDDAT ;ALL BITS OK ?
061266 001401      BEQ     69$ ;BR IF OK FROM PORT A.
061270 104007      EMT     7
061272 013737 001172 001126 69$: MOV     $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
061300 013737 001226 001240  MOV     PORTB, PTNBR ;CHANGE PORT NUMBER
061306 023737 001124 001126  CMP     $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
061314 001401      BEQ     70$ ;BR IF OK
061316 104007      EMT     7
061320 000240      70$: NOP

```

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

061322 023737 001276 001272      CMP     TIMES, TIMEBP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
061330 003004      BGT     4$ ;BR IF GREATER
061332 023737 001276 001274      CMP     TIMES, TIMEBM ;MEASURED TIME LESS THAN -25% TOLERANCE
061340 002001      BGE     .+4 ;BR IF NOT
061342      4$:
061342 104025      EMT     25
061344 000004      SCOPE ;LOOP ?

```

1593  
1614  
1615

\*\*\*\*\*  
 ;\*TEST 46 PORT 'A' TIMEOUT/RELEASE TEST



- \* VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
- \* A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMDS.
- \* B. SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'A'.
- \* C. ISSUE A RELEASE COMMAND FROM PORT 'B'. VERIFY THAT THE DRIVE HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT SET FOR PORT 'B'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'A'.
- \* D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS BEEN RELEASED.
- \* E. RELEASE THE DRIVE THROUGH PORT 'A'. VERIFY THAT THE DRIVE RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.

\*\*\*\*\*

```
TST46:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
BEQ      2$          ;BR IF NOT
BPL      1$          ;BR IF JUST ENTERED TEST
JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
2$:      MOV      #TEST46,$LPADR ;SETUP SCOPE LOOP ADDRESS
          MOV      #TEST46,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST46:
          MOVB     #46,$STSTM      ;MOVE #46 TO TEST NUMBER
          MOV      #STACK,$SP      ;LOAD THE STACK POINTER
          MOV      #2,$TIMES       ;;DO 2. ITERATIONS
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```
061424 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT #A
061432 005060 000012              CLR      RMDS(R0)        ;SEIZE THE DRIVE
061436 012760 000011 000000      MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
061444 012760 000013 000000      MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE
061452 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT #B
061460 005060 000012              CLR      RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
061464 012760 000011 000000      MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
061472 012760 000013 000000      MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE
```

;SEIZE THE DRIVE THROUGH PORT B

```
061500 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
061506 013737 001226 001242      MOV      PORTB, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
061514 005060 000012              CLR      RMDS(R0)        ;WRITE RMDS
061520 013737 001224 001244      MOV      PORTA, OPPRT    ;'OPPOSITE' PORT ADDRESS
061526 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
061534 013737 001224 001240      MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

;SET REQUEST THROUGH PORT A

```
061542 005060 000012              CLR      RMDS(R0)        ;SET REQUEST FOR PORT A
061546 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
```

1616  
1660

```

061554 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
;RELEASE THE DRIVE THROUGH PORT B

061562 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE DRIVE THROUGH PORT B
;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)

061570 013737 001264 001260      MOV      TIMEAP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
;VERIFY THAT THE DRIVE IS SEIZED BY PORT A

061576 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
061602 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
061610 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
061616 060037 001122      ADD      R0,$BDADR ;ADD RH/RM BASE ADDRESS
061622 005037 001124      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
061626 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
061634 001403      BEQ      66$ ;BR IF OK
061636 104031      EMT      31
061640 005137 001250      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
061644 000240      NOP
061646 005737 001250      TST      CKERR ;REGISTER OK ?
061652 001402      BEQ      .+6 ;BR IF OK
061654 000137 062230      JMP      1$ ;BYPASS REST OF TEST IF NOT

;WAIT FOR THE TIMER TO RELEASE THE DRIVE
061660 005737 001260      JST      WATCH ;WATCH EQUAL ZERO ?
061664 001375      BNE      .-4 ;BR IF NOT

;CONFIRM THAT THE DRIVE HAS TIMED OUT
;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

061666 005037 001254      CLR      RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
061672 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
061700 060037 001122      ADD      R0,$BDADR ;ADD THE I/O BASE ADDRESS
061704 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
061712 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
061720 016037 000012 001170      MOV      RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
061726 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
061734 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
061742 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061750 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
061756 016037 000012 001172      MOV      RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061764 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
061772 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
062000 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062006 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
062014 001006      BNE      68$ ;BR IF NOT
062016 005737 001164      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
062022 001045      BNE      70$ ;BR IF NOT
062024 104046      EMT      46
062026 000137 062226      JMP      72$ ;BYPASS THE REST OF THE CHECKS
062032 013737 001170 001126 68$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
062040 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
  
```



```

062046 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
062054 005737 001164              TST    $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
062060 001414                      BEQ    69$             ;BR IF ZERO
062062 013737 001224 001240      MOV    PORTA, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062070 013737 001172 001126      MOV    $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
062076 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
062104 005737 001166              TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
062110 001012                      BNE   70$             ;BR IF NOT
062112 012737 177777 001254 69$:  MOV    #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
062120 012760 000011 000000      MOV    #11, RMCS1(R0) ;CLEAR THE DRIVE
062126 012760 000013 000000      MOV    #13, RMCS1(R0) ;RELEASE THE DRIVE
062134 104035                      EMT    35
062136 013737 001170 001126 70$:  MOV    $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
062144 013737 001224 001240      MOV    PORTA, PTNBR    ;CHANGE PORT NUMBER
062152 042737 100000 001126      BIC    #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
062160 023737 001124 001126      CMP    $GDDAT, $BDDAT ;ALL BITS OK ?
062166 001401                      BEQ    71$             ;BR IF OK FROM PORT A.
062170 104007                      EMT    7
062172 013737 001172 001126 71$:  MOV    $TMP3, $BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
062200 013737 001226 001240      MOV    PORTB, PTNBR    ;CHANGE PORT NUMBER
062206 042737 100000 001126      BIC    #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
062214 023737 001124 001126      CMP    $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
062222 001401                      BEQ    72$             ;BR IF OK
062224 104007                      EMT    7
062226 000240                      NOP
062230 000004                      1$:   SCOPE           ;LOOP ?
  
```

1661  
1679  
1680

```

*****
*TEST 47      PORT 'B' TIMEOUT/RELEASE TEST
*
*VERIFY THAT THE TIMEOUT ONE-SHOT IS TRIGGERED WHEN THE DRIVE
*SWITCHES PORTS AND SEIZING PORT PERFORMS NO REGISTER ACCESSES.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMDS.
*  B. SET PORT REQUEST BY WRITING 0'S INTO RMDS FROM PORT 'B'.
*  C. ISSUE A RELEASE COMMAND FROM PORT 'A'. VERIFY THAT THE DRIVE
*     HAS SWITCHED TO THE OTHER PORT AND THAT THE 'ATA' BIT DID NOT
*     SET FOR PORT 'A'. REGISTERS WILL NOT BE CHECKED THROUGH PORT 'B'.
*  D. WAIT THE TIMEOUT INTERVAL + 25%. VERIFY THAT THE DRIVE HAS
*     BEEN RELEASED.
*
*****
  
```

```

062232 005737 001300      TST47: TST    KYBCTL       ;PERFORMING ONLY SINGLE TEST ?
062232 001406                      BEQ    2$             ;BR IF NOT
062240 100002                      BPL    1$             ;BR IF JUST ENTERED TEST
062242 000137 003110      JMP    EXEC           ;RETURN & GET NEXT TEST NUMBER
062246 012737 177777 001300 1$:   MOV    #-1, KYBCTL    ;SET SINGLE TEST INDICATOR
062254 012737 062270 001106 2$:   MOV    #TEST47, $LPADR ;SETUP SCOPE LOOP ADDRESS
062262 012737 062270 001110      MOV    #TEST47, $LPERR ;SETUP ERROR LOOP ADDRESS
062270                                TEST47:
062270 112737 000047 001102      MOVB   #47, $TSTNM    ;MOVE #47 TO TEST NUMBER
062276 012706 001100      MOV    #STACK, SP    ;LOAD THE STACK POINTER
  
```



1681  
1682

```

062302 012737 000002 001176      MOV      #2, $TIMES      ;;DO 2. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

062310 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT #A
062316 005060 000012 000000      CLR      RMDS(R0)       ;SEIZE THE DRIVE
062322 012760 000011 000000      MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
062330 012760 000013 000000      MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE
062336 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT #B
062344 005060 000012 000000      CLR      RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
062350 012760 000011 000000      MOV      #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
062356 012760 000013 000000      MOV      #13, RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

062364 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
062372 013737 001224 001242      MOV      PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
062400 005060 000012 000000      CLR      RMDS(R0)       ;WRITE RMDS
062404 013737 001226 001244      MOV      PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
062412 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
062420 013737 001226 001240      MOV      PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET REQUEST THROUGH PORT B

062426 005060 000012 000000      CLR      RMDS(R0)       ;SET REQUEST FOR PORT B
062432 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
062440 013737 001224 001240      MOV      PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;RELEASE THE DRIVE THROUGH PORT A

062446 012760 000013 000000      MOV      #13, RMCS1(R0)  ;RELEASE DRIVE THROUGH PORT A

;WAIT THE MEASURED TIMEOUT FOR THE PORT (+ 25%)

062454 013737 001272 001260      MOV      TIMEBP, WATCH   ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%

;VERIFY THAT THE DRIVE IS SEIZED BY PORT B

062462 005037 001250 000000      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
062466 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
062474 012737 000012 001122      MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
062502 060037 001122 000000      ADD      R0, $BDADR      ;ADD RH/RM BASE ADDRESS
062506 005037 001124 000000      CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
062512 023737 001124 001126      CMP      $GDDAT, $BDDAT  ;IS THE REGISTER OK ?
062520 001403 000000 000000      BEQ      66$            ;BR IF OK
062522 104031 000000 000000      EMT      31             ;
062524 005137 001250 000000      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
062530 000240 000000 000000      66$:  NOP               ;
062532 005737 001250 000000      TST      CKERR          ;REGISTER OK ?
062536 001402 000000 000000      BEQ      +6             ;BR IF OK
062540 000137 063114 000000      JMP      1$             ;BYPASS REST OF TEST IF NOT

;WAIT FOR THE TIMER TO RELEASE THE DRIVE

062544 005737 001260 000000      TST      WATCH         ;WATCH EQUAL ZERO ?
062550 001375 000000 000000      BNE     -4             ;BR IF NOT
    
```



;CONFIRM THAT THE DRIVE HAS TIMED OUT

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

062552 005037 001254          CLR      RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
062556 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
062564 060037 001122          ADD      RO,$BDADR   ;ADD THE I/O BASE ADDRESS
062570 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
062576 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
062604 016037 000012 001170  MOV      RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
062612 042737 024001 001170  BIC     #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
062620 013737 001170 001164  MOV      $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
062626 042737 100100 001164  BIC     #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062634 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
062642 016037 000012 001172  MOV      RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
062650 042737 024001 001172  BIC     #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
062656 013737 001172 001166  MOV      $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
062664 042737 100100 001166  BIC     #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062672 023737 001164 001166  CMP     $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
062700 001006          BNE     68$         ;BR IF NOT
062702 005737 001164          TST     $TMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
062706 001045          BNE     70$         ;BR IF NOT
062710 104046          EMT     46
062712 000137 063112          JMP     72$         ;BYPASS THE REST OF THE CHECKS
062716 013737 001170 001126 68$:  MOV     $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
062724 013737 001226 001240  MOV     PORTB, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062732 113760 001226 000010  MOVVB   PORTB, RMCS2(RO) ;SELECT PORT B.
062740 005737 001164          TST     $TMP0       ;SEE IF STATUS EQ 0 FROM PORT A.
062744 001414          BEQ     69$         ;BR IF ZERO
062746 013737 001224 001240  MOV     PORTA, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062754 013737 001172 001126  MOV     $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
062762 113760 001224 000010  MOVVB   PORTA, RMCS2(RO) ;SELECT PORT A.
062770 005737 001166          TST     $TMP1       ;SEE IF STATUS EQ ZERO FROM PORT B.
062774 001012          BNE     70$         ;BR IF NOT
062776 012737 177777 001254 69$:  MOV     #-1, RELERR  ;SET 'RELEASE ERROR' INDICATOR
063004 012760 000011 000000  MOV     #11, RMCS1(RO) ;CLEAR THE DRIVE
063012 012760 000013 000000  MOV     #13, RMCS1(RO) ;RELEASE THE DRIVE
063020 104035          EMT     35
063022 013737 001170 001126 70$:  MOV     $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
063030 013737 001224 001240  MOV     PORTA, PTNBR  ;CHANGE PORT NUMBER
063036 042737 100000 001126  BIC     #ATA, $BDDAT  ;DON'T CHECK THE ATTN BIT
063044 023737 001124 001126  CMP     $GDDAT, $BDDAT ;ALL BITS OK ?
063052 001401          BEQ     71$         ;BR IF OK FROM PORT A.
063054 104007          EMT     7
063056 013737 001172 001126 71$:  MOV     $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
063064 013737 001226 001240  MOV     PORTB, PTNBR  ;CHANGE PORT NUMBER
063072 042737 100000 001126  BIC     #ATA, $BDDAT  ;DON'T CHECK THE ATTN BIT
063100 023737 001124 001126  CMP     $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
063106 001401          BEQ     72$         ;BR IF OK
063110 104007          EMT     7
063112 000240          NOP
063114 000004          1$:  SCOPE          ;LOOP ?
    
```

1683  
1708  
1709

\*\*\*\*\*  
 ;\*TEST 50 PORT 'A' SEIZE ACCESS TEST





```

063320 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063326 004737 064166              JSR     PC,TST50B  ;CHECK THE REGISTERS THROUGH PORT B
063332 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
063340 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063346 005060 000042              CLR     RMER2(RO)  ;CLEAR RMER2 ON PORT A
063352 005060 000014              CLR     RMER1(RO)  ;CLEAR RMER1 ON PORT A
063356 013760 001236 000016      MOV     ASR1,RMAS(RO) ;CLEAR THE ATTENTION BIT FOR PORT A
063364 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B
063372 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063400 012760 177777 000014      MOV     #-1,RMER1(RO) ;LOAD 1'S INTO RMER1 THROUGH PORT B
063406 012760 177777 000042      MOV     #-1,RMER2(RO) ;LOAD 1'S INTO RMER2 THROUGH PORT B
063414 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
063422 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063430 004737 064166              JSR     PC,TST50B  ;CHECK THE REGISTERS THROUGH PORT A

```

;RELEASE THE DRIVE FROM PORT A

```

063434 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
063442 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063450 012760 000013 000000      MOV     #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

```

;VERIFY THAT DRIVE IS SEIZED BY PORT B WHEN RELEASED BY PORT A

```

063456 005037 001254              CLR     RELERR     ;CLEAR 'RELEASE ERROR' INDICATOR
063462 012737 111700 001124      MOV     #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
063470 012737 000012 001122      MOV     #RMDS,$BDADR ;REGISTER ADDRESS INCREMENT
063476 060037 001122              ADD     RO,$BDADR  ;REGISTER BASE ADDRESS FOR TYPEOUT
063502 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B
063510 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063516 016037 000012 001164      MOV     RMDS(RO),$TMP0 ;READ STATUS REGISTER FROM PORT B
063524 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A
063532 013737 001224 001240      MOV     PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063540 016037 000012 001126      MOV     RMDS(RO),$BDDAT ;DRIVE STATUS FROM PORT A
063546 001404              BEQ     66$        ;BR IF STATUS FROM PORT A ZERO
063550 005737 001164              TST     $TMP0      ;IS STATUS FROM PORT B ZERO ?
063554 001401              BEQ     66$        ;BR IF ZERO
063556 104031              EMT     31
063560 013737 001164 001126 66$:  MOV     $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
063566 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
063574 023737 001124 001126      CMP     $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
063602 001401              BEQ     67$        ;BR IF OK
063604 104027              EMT     27
063606 000240 67$:  NOP

```

;RELEASE THE DRIVE FROM PORT B

```

063610 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B
063616 013737 001226 001240      MOV     PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063624 012760 000013 000000      MOV     #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

063632 005037 001254              CLR     RELERR     ;CLEAR THE 'RELEASE ERROR ' INDICATOR
063636 012737 000012 001122      MOV     #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
063644 060037 001122              ADD     RO,$BDADR  ;ADD THE I/O BASE ADDRESS
063650 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
063656 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.

```

```

063664 016037 000012 001170      MOV      RMDS(R0), $TMP2      ;GET THE DRIVE STATUS REGISTER FROM PORT A.
063672 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2  ;CLEAR DONT CARES
063700 013737 001170 001164      MOV      $TMP2, $TMP0        ;COPY IT INTO '$TMP0'
063706 042737 100100 001164      BIC      #ATA!VV, $TMP0       ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063714 113760 001226 000010      MOV      PORTB, RMCS2(R0)    ;SELECT PORT B.
063722 016037 000012 001172      MOV      RMDS(R0), $TMP3     ;GET THE DRIVE STATUS REGISTER FROM PORT B.
063730 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3  ;CLEAR DONT CARES
063736 013737 001172 001166      MOV      $TMP3, $TMP1        ;COPY IT INTO '$TMP1'
063744 042737 100100 001166      BIC      #ATA!VV, $TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063752 023737 001164 001166      CMP      $TMP0, $TMP1        ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
063760 001006 001164 001166      BNE      68$                 ;BR IF NOT
063762 005737 001164 001166      TST      $TMP0               ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
063766 001045 001164 001166      BNE      70$                 ;BR IF NOT
063770 104046 001164 001166      EMT      46
063772 000137 064156 001166      JMP      72$                 ;BYPASS THE REST OF THE CHECKS
063776 013737 001170 001126 68$:  MOV      $TMP2, $BDDAT       ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
064004 013737 001226 001240      MOV      PORTB, PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
064012 113760 001226 000010      MOV      PORTB, RMCS2(R0)    ;SELECT PORT B.
064020 005737 001164 001166      TST      $TMP0               ;SEE IF STATUS EQ 0 FROM PORT A.
064024 001414 001164 001166      BEQ      69$                 ;BR IF ZERO
064026 013737 001224 001240      MOV      PORTA, PTNBR        ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
064034 013737 001172 001126      MOV      $TMP3, $BDDAT       ;'BAD DATA' FOR ERROR TYPE OUT
064042 113760 001224 000010      MOV      PORTA, RMCS2(R0)    ;SELECT PORT A.
064050 005737 001166 001166      TST      $TMP1               ;SEE IF STATUS EQ ZERO FROM PORT B.
064054 001012 001166 001166      BNE      70$                 ;BR IF NOT
064056 012737 177777 001254 69$:  MOV      #-1, RELERR         ;SET 'RELEASE ERROR' INDICATOR
064064 012760 000011 000000      MOV      #11, RMCS1(R0)      ;CLEAR THE DRIVE
064072 012760 000013 000000      MOV      #13, RMCS1(R0)      ;RELEASE THE DRIVE
064100 104026 001166 001166      EMT      26
064102 013737 001170 001126 70$:  MOV      $TMP2, $BDDAT       ;LOOK FOR BIT FAILURES WHEN RMDS READ
064110 013737 001224 001240      MOV      PORTA, PTNBR        ;CHANGE PORT NUMBER
064116 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;ALL BITS OK ?
064124 001401 001124 001126      BEQ      71$                 ;BR IF OK FROM PORT A.
064126 104007 001124 001126      EMT      7
064130 013737 001172 001126 71$:  MOV      $TMP3, $BDDAT       ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
064136 013737 001226 001240      MOV      PORTB, PTNBR        ;CHANGE PORT NUMBER
064144 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;SEE IF READ OK FROM PORT B.
064152 001401 001124 001126      BEQ      72$                 ;BR IF OK
064154 104007 001124 001126      EMT      7
064156 000240 001124 001126 72$:  NOP
064160 000004 001124 001126      SCOPE
1750 064162 000137 064410      JMP      TST51                ;LOOP ?
                                ;GO TO THE NEXT TEST

```

;CHECK THE REGISTERS ON THE SELECTED PORT

```

064166 005037 001250 001126      TST50B: CLR      CKERR                ;CLEAR THE 'CHECK ERROR' INDICATOR
064172 016037 000014 001126      MOV      RMER1(R0), $BDDAT   ;GET CONTENTS OF RMER1
064200 012737 000014 001122      MOV      #RMER1, $BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
064206 060037 001122 001122      ADD      R0, $BDADR          ;ADD RH/RM BASE ADDRESS
064212 005037 001124 001126      CLR      $GDDAT              ;WHAT REGISTER SHOULD BE
064216 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;IS THE REGISTER OK ?
064224 001403 001124 001126      BEQ      64$                 ;BR IF OK
064226 104006 001124 001126      EMT      6
064230 005137 001250 001126      COM      CKERR                ;SET THE REGISTER COMPARE ERROR INDICATOR
064234 016037 000000 001126 64$:  MOV      RMCS1(R0), $BDDAT   ;GET THE CONTENTS OF RHCS1
064242 012737 000000 001122      MOV      #RMCS1, $BDADR      ;FORM ADDRESS OF REGISTER

```



```

064250 060037 001122          ADD    RO,$BDADR      ;ADDRESS BASE
064254 032737 020000 001126    BIT    #MCPE,$BDDAT   ;IS 'MCPE' SET ?
064262 001404                BEQ    65$            ;BR IF NOT
064264 104011                EMT    11
064266 012760 040000 000000    MOV    #TRE,RMCS1(RO) ;CLEAR 'MCPE'
064274 000240                NOP
064276 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
064302 016037 000042 001126    MOV    RMER2(RO),$BDDAT ;GET CONTENTS OF RMER2
064310 012737 000042 001122    MOV    #RMER2,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
064316 060037 001122          ADD    RO,$BDADR      ;ADD RH/RM BASE ADDRESS
064322 005037 001124          CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
064326 023737 001124 001126    CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
064334 001403                BEQ    66$            ;BR IF OK
064336 104006                EMT    6
064340 005137 001250          COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
064344 016037 000000 001126    MOV    RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
064352 012737 000000 001122    MOV    #RMCS1,$BDADR  ;FORM ADDRESS OF REGISTER
064360 060037 001122          ADD    RO,$BDADR      ;ADDRESS BASE
064364 032737 020000 001126    BIT    #MCPE,$BDDAT   ;IS 'MCPE' SET ?
064372 001404                BEQ    67$            ;BR IF NOT
064374 104011                EMT    11
064376 012760 040000 000000    MOV    #TRE,RMCS1(RO) ;CLEAR 'MCPE'
064404 000240                NOP
064406 000207                RTS                    ;RETURN
    
```

1751  
 1776  
 1777

```

*****
*TEST 51      PORT 'B' SEIZE ACCESS TEST
*
*VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.
*
*  A. SEIZE THE DRIVE THROUGH PORT 'B' BY WRITING 0'S INTO RMD5.
*
*  B. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'B'.
*
*  C. READ RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT PORT
*     'A' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  D. CLEAR RMER1, RMER2 THROUGH PORT 'B'.
*
*  E. WRITE 1'S INTO RMER1, RMER2 THROUGH PORT 'A'. VERIFY THAT
*     PORT 'B' SEES 0'S FROM EACH OF THESE REGISTERS.
*
*  F. RELEASE THE DRIVE THROUGH PORT 'B'. VERIFY THAT THE DRIVE HAS
*     SWITCHED TO PORT 'A' AND THAT THE ATTENTION BIT FOR PORT 'A' IS
*     SET AND THE ATTENTION BIT FOR PORT 'B' IS NOT SET.
*
*  G. ISSUE A RELEASE COMMAND THROUGH PORT 'A'. VERIFY THAT THE DRIVE
*     RETURNED TO NEUTRAL AND THAT NEITHER ATTENTION BIT IS SET.
*
*****
    
```

```

064410 005737 001300          TST    KYBCTL        ;PERFORMING ONLY SINGLE TEST ?
064414 001406                BEQ    2$            ;BR IF NOT
064416 100002                BPL    1$            ;BR IF JUST ENTERED TEST
064420 000137 003110          JMP    EXEC          ;RETURN & GET NEXT TEST NUMBER
064424 012737 177777 001300    MOV    #-1,KYBCTL    ;SET SINGLE TEST INDICATOR
    
```

1778  
1779

```

064432 012737 064446 001106 2$: MOV #TEST51,$LPADR ;SETUP SCOPE LOOP ADDRESS
064440 012737 064446 001110 MOV #TEST51,$LPERR ;SETUP ERROR LOOP ADDRESS
064446 TEST51:
064446 112737 000051 001102 MOVB #51,$TSTNM ;MOVE #51 TO TEST NUMBER
064454 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
064460 012737 000031 001176 MOV #25.,$TIMES ;;DO 25. ITERATIONS
    
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

064466 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT #A
064474 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE
064500 012760 000011 000000 MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
064506 012760 000013 000000 MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
064514 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT #B
064522 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
064526 012760 000011 000000 MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
064534 012760 000013 000000 MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
    
```

;SEIZE THE DRIVE THROUGH PORT B

```

064542 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
064550 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
064556 005060 000012 CLR RMDS(R0) ;WRITE RMDS
064562 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
064570 012760 177777 000014 MOV #-1, RMER1(R0) ;LOAD 1'S INTO RMER1 THROUGH PORT B
064576 012760 177777 000042 MOV #-1, RMER2(R0) ;LOAD 1'S INTO RMER2 THROUGH PORT B
064604 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
064612 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064620 004737 065460 JSR PC, TST51B ;CHECK THE REGISTERS THROUGH PORT A
064624 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
064632 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064640 005060 000042 CLR RMER2(R0) ;CLEAR RMER2 ON PORT B
064644 005060 000014 CLR RMER1(R0) ;CLEAR RMER1 ON PORT B
064650 013760 001236 000016 MOV ASR1, RMAS(R0) ;CLEAR THE ATTENTION BIT FOR PORT B
064656 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
064664 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064672 012760 177777 000014 MOV #-1, RMER1(R0) ;LOAD 1'S INTO RMER1 THROUGH PORT A
064700 012760 177777 000042 MOV #-1, RMER2(R0) ;LOAD 1'S INTO RMER2 THROUGH PORT A
064706 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
064714 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064722 004737 065460 JSR PC, TST51B ;CHECK THE REGISTERS THROUGH PORT B
    
```

;RELEASE THE DRIVE FROM PORT B

```

064726 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
064734 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064742 012760 000013 000000 MOV #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
    
```

;VERIFY THAT DRIVE IS SEIZED BY PORT A WHEN RELEASED BY PORT B

```

064750 005037 001254 CLR RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
064754 012737 111700 001124 MOV #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
064762 012737 000012 001122 MOV #RMDS,$BDADR ;REGISTER ADDRESS INCREMENT
064770 060037 001122 ADD R0,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
064774 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
065002 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```



```

065010 016037 000012 001164      MOV      RMDS(R0), $TMP0      ;READ STATUS REGISTER FROM PORT A
065016 113760 001226 000010      MOV      PORTB, RMCS2(R0)    ;SELECT PORT B
065024 013737 001226 001240      MOV      PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065032 016037 000012 001126      MOV      RMDS(R0), $BDDAT    ;DRIVE STATUS FROM PORT B
065040 001404      BEQ      66$                ;BR IF STATUS FROM PORT B ZERO
065042 005737 001164      TST      $TMP0              ;IS STATUS FROM PORT A ZERO ?
065046 001401      BEQ      66$                ;BR IF ZERO
065050 104031      EMT      31
065052 013737 001164 001126 66$:  MOV      $TMP0, $BDDAT      ;CHECK STATUS FROM PORT A
065060 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT ADDRESS FOR TYPEOUT
065066 023737 001124 001126      CMP      $GDDAT, $BDDAT    ;COMPARE WITH CONSTANT
065074 001401      BEQ      67$                ;BR IF OK
065076 104027      EMT      27
065100 000240      NOP

;RELEASE THE DRIVE FROM PORT A

065102 113760 001224 000010      MOV      PORTA, RMCS2(R0)   ;SELECT PORT A
065110 013737 001224 001240      MOV      PORTA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065116 012760 000013 000000      MOV      #13, RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

065124 005037 001254      CLR      RELERR             ;CLEAR THE 'RELEASE ERROR ' INDICATOR
065130 012737 000012 001122      MOV      #RMDS, $BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
065136 060037 001122      ADD      R0, $BDADR        ;ADD THE I/O BASE ADDRESS
065142 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
065150 113760 001224 000010      MOV      PORTA, RMCS2(R0)  ;SELECT PORT A.
065156 016037 000012 001170      MOV      RMDS(R0), $TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
065164 042737 024001 001170      BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
065172 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
065200 042737 100100 001164      BIC      #ATA!VV, $TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065206 113760 001226 000010      MOV      PORTB, RMCS2(R0)  ;SELECT PORT B.
065214 016037 000012 001172      MOV      RMDS(R0), $TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
065222 042737 024001 001172      BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
065230 013737 001172 001166      MOV      $TMP3, $TMP1     ;COPY IT INTO '$TMP1'
065236 042737 100100 001166      BIC      #ATA!VV, $TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065244 023737 001164 001166      CMP      $TMP0, $TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
065252 001006      BNE      68$                ;BR IF NOT
065254 005737 001164      TST      $TMP0              ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
065260 001045      BNE      70$                ;BR IF NOT
065262 104046      EMT      46
065264 000137 065450      JMP      72$                ;BYPASS THE REST OF THE CHECKS
065270 013737 001170 001126 68$:  MOV      $TMP2, $BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
065276 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065304 113760 001226 000010      MOV      PORTB, RMCS2(R0)  ;SELECT PORT B.
065312 005737 001164      TST      $TMP0              ;SEE IF STATUS EQ 0 FROM PORT A.
065316 001414      BEQ      69$                ;BR IF ZERO
065320 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065326 013737 001172 001126      MOV      $TMP3, $BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
065334 113760 001224 000010      MOV      PORTA, RMCS2(R0)  ;SELECT PORT A.
065342 005737 001166      TST      $TMP1              ;SEE IF STATUS EQ ZERO FROM PORT B.
065346 001012      BNE      70$                ;BR IF NOT
065350 012737 177777 001254 69$:  MOV      #-1, RELERR       ;SET 'RELEASE ERROR' INDICATOR
065356 012760 000011 000000      MOV      #11, RMCS1(R0)    ;CLEAR THE DRIVE
065364 012760 000013 000000      MOV      #13, RMCS1(R0)    ;RELEASE THE DRIVE
065372 104026      EMT      26
  
```

```

065374 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
065402 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
065410 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
065416 001401 BEQ 71$ ;BR IF OK FROM PORT A.
065420 104007 EMT 7
065422 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
065430 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
065436 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
065444 001401 BEQ 72$ ;BR IF OK
065446 104007 EMT 7
065450 000240 72$: NOP
065452 000004 SCOPE ;LOOP ?
1780 065454 000137 065702 JMP TST52 ;GO TO THE NEXT TEST
    
```

;CHECK THE REGISTERS ON THE SELECTED PORT

```

065460 TST51B:
065460 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
065464 016037 000014 001126 MOV RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
065472 012737 000014 001122 MOV #RMER1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
065500 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
065504 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
065510 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
065516 001403 BEQ 64$ ;BR IF OK
065520 104006 EMT 6
065522 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
065526 016037 000000 001126 64$: MOV RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
065534 012737 000000 001122 MOV #RMCS1,$BDADR ;FORM ADDRESS OF REGISTER
065542 060037 001122 ADD RO,$BDADR ;ADDRESS BASE
065546 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
065554 001404 BEQ 65$ ;BR IF NOT
065556 104011 EMT 11
065560 012760 040000 000000 MOV #TRE,RMCS1(RO) ;CLEAR 'MCPE'
065566 000240 65$: NOP
065570 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
065574 016037 000042 001126 MOV RMER2(RO),$BDDAT ;GET CONTENTS OF RMER2
065602 012737 000042 001122 MOV #RMER2,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
065610 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
065614 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
065620 023737 001124 001126 CMP $GDDAT,$BDDAT ;IS THE REGISTER OK ?
065626 001403 BEQ 66$ ;BR IF OK
065630 104006 EMT 6
065632 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
065636 016037 000000 001126 66$: MOV RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
065644 012737 000000 001122 MOV #RMCS1,$BDADR ;FORM ADDRESS OF REGISTER
065652 060037 001122 ADD RO,$BDADR ;ADDRESS BASE
065656 032737 020000 001126 BIT #MCPE,$BDDAT ;IS 'MCPE' SET ?
065664 001404 BEQ 67$ ;BR IF NOT
065666 104011 EMT 11
065670 012760 040000 000000 MOV #TRE,RMCS1(RO) ;CLEAR 'MCPE'
065676 000240 67$: NOP
065700 000207 RTS PC ;RETURN
    
```

1781  
1782  
1783  
1784

065702 000004

```

;*****
;PUT NEWTEST HERE
;*****
TST52: SCOPE
    
```



1785  
 1791  
 1792

.SBTTL END OF PASS ROUTINE

```

:*****
:*INCREMENT THE PASS NUMBER ($PASS)
:*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
:*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 TST1AA
    
```

```

065704      065704 005737 001300      $EOP:      TST      KYBCTL      ;ENTERED TEST VIA KEYBOARD COMMAND ?
065710      065710 001402                BEQ      .+6      ;BR IF NOT
065712      065712 000137 003110      JMP      EXEC     ;RETURN TO KEYBOARD CONTROL
065716      065716 005037 001102      CLR      $TSTNM  ;:ZERO THE TEST NUMBER
065722      065722 005037 001176      CLR      $TIMES  ;:ZERO THE NUMBER OF ITERATIONS
065726      065726 005237 001100      INC      $PASS   ;:INCREMENT THE PASS NUMBER
065732      065732 042737 100000 001100  BIC      #100000,$PASS ;:DON'T ALLOW A NEG. NUMBER
065740      065740 005327                DEC      (PC)+   ;:LOOP?
065742      065742 000001      $EOPCT: .WORD 1
065744      065744 003063      BGT      $DOAGN  ;:YES
065746      065746 012737      MOV      (PC)+,@(PC)+ ;:RESTORE COUNTER
065750      065750 000001      $ENDCT: .WORD 1
065752      065752 065742      $EOPCT
065754      065754 104401 065762      TYPE     ,65$    ;:TYPE ASCIZ STRING
065760      065760 000407      BR       64$     ;:GET OVER THE ASCIZ
;:65$: .ASCIZ <12><15>/END PASS #/
64$:
066000      066000 013746 001100      MOV      $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
;:TYPE PASS NUMBER
066004      066004 104405      TYPDS   ;:GO TYPE--DECIMAL ASCII WITH SIGN
066006      066006 104401 066014      TYPE     ,67$    ;:TYPE ASCIZ STRING
066012      066012 000421      BR       66$     ;:GET OVER THE ASCIZ
;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
066056      066056 013746 001112      MOV      $ERTTL,-(SP) ;:SAVE $ERTTL FOR TYPEOUT
;:TOTAL NUMBER OF ERRORS
066062      066062 104405      TYPDS   ;:GO TYPE--DECIMAL ASCII WITH SIGN
066064      066064 104401 001207      TYPE     , $CRLF ;:TYPE CARRIAGE RETURN, LINE FEED
066070      066070 005037 001112      CLR      $ERTTL  ;:CLEAR ERROR TOTAL
066074      066074 013700 000042  $GET42: MOV      @#42,R0  ;:GET MONITOR ADDRESS
066100      066100 001405      BEQ      $DOAGN  ;:BRANCH IF NO MONITOR
066102      066102 000005      RESET   ;:CLEAR THE WORLD
066104      066104 004710      $ENDAD: JSR      PC,(R0) ;:GO TO MONITOR
066106      066106 000240      NOP     ;:SAVE ROOM
066110      066110 000240      NOP     ;:FOR
066112      066112 000240      NOP     ;:ACT11
066114      066114 000137      $DOAGN: JMP      @(PC)+   ;:RETURN
066116      066116 003372      $RTNAD: .WORD  TST1AA
066120      066120      377      377      000  $ENULL: .BYTE  -1,-1,0 ;:NULL CHARACTER STRING
;:EVEN
    
```

1793

```

1          .SBTTL SUBROUTINES
2          :*****
3          .SBTTL CLOCK SUBROUTINES
4
5          ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
6          ;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED
7
8 066124 012737 066174 000004 CKCLK:  MOV    #CKCLK1,@#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
9 066132 005037 000006          CLR    @#ERRVEC+2 ;NEW PSW
10 066136 005777 113050          TST    @SLKCSR ;CHECK FOR KW11-P
11 066142 013701 001216          MOV    $LPVEC,R1 ;KW11-P VECTOR ADDRESS
12 066146 012721 066256          MOV    #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
13 066152 012711 000300          MOV    #300,(R1) ;PSW - PRI 6
14 066156 012777 177777 113030          MOV    #-1,@SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
15 066164 012777 000135 113020          MOV    #135,@SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
16 066172 000425          BR     CKCLK3
17 066174 062706 000004          CKCLK1: ADD   #4,SP ;RESTORE THE STACK POINTER
18 066200 012737 066236 000004          MOV    #CKCLK2,@#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
19 066206 005777 113006          TST    @SLKS ;LOOK FOR KW11-L
20 066212 013701 001222          MOV    $LLVEC,R1 ;KW11-L VECTOR ADDRESS
21 066216 012721 066256          MOV    #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
22 066222 012711 000300          MOV    #300,(R1) ;PSW - PRI 6
23 066226 012777 000100 112764          MOV    #100,@SLKS ;SET KW11-L INTERRUPT
24 066234 000404          BR     CKCLK3
25 066236 062706 000004          CKCLK2: ADD   #4,SP ;RESTORE THE STACK POINTER
26 066242 062716 000002          ADD   #2,(SP) ;INCREMENT RETURN, NO CLOCK
27 066246 012737 000006 000004          CKCLK3: MOV   #6,@#ERRVEC ;RESTORE THE ERROR VECTOR
28 066254 000207          RTS    PC
29
30          ;ROUTINE TO COUNT CLOCK TICKS
31
32 066256 062737 000021 001256          CLOCK: ADD   #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
33 066264 103003          BCC   1$ ;BRANCH IF NO OVERFLOW
34 066266 012737 177777 001256          MOV   #-1,TIME ;OVERFLOW - RESTORE MAXIMUM COUNT
35 066274 005737 001260          1$:  TST   WATCH ;IS WATCH ALREADY ZERO ?
36 066300 001406          BEQ   2$ ;BR IF IT IS
37 066302 162737 000021 001260          SUB   #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
38 066310 100002          BPL   2$ ;BR IF NOT MINUS
39 066312 005037 001260          CLR   WATCH ;CLEAR WATCH DOG COUNTER
40 066316 000002          2$:  RTI   ;RETURN
41
42          ;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES
43
44 066320 162706 000004          TOLER: SUB   #4,SP ;SETUP STACK
45 066324 016616 000004          MOV   4(SP),(SP) ;SAVE STACK
46 066330 013546          MOV   @R5+,-(SP) ;GET TIME VALUE
47 066332 011666 000004          MOV   (SP),4(SP) ;MOVE TIME VALUE
48 066336 011666 000006          MOV   (SP),6(SP) ;MOVE VALUE AGAIN
49 066342 006216          ASR   (SP) ;DIVIDE BY 2
50 066344 006216          ASR   (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
51 066346 061666 000004          ADD   (SP),4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
52 066352 162666 000004          SUB   (SP)+,4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
53 066356 000205          RTS   R5 ;RETURN WITH TOLERANCES ON THE STACK
54

```



1

.SBTTL SCOPE HANDLER ROUTINE

```

*****
*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
*SW14=1      LOOP ON TEST
*SW11=1      INHIBIT ITERATIONS
*SW09=1      LOOP ON ERROR
*CALL
*          SCOPE          ;;SCOPE=IOT
    
```

```

066360          $SCOPE:
066360 104407          CKSWR
066362 032777 040000 112550 1$: BIT #BIT14,@SWR ;;TEST FOR CHANGE IN SOFT-SWR
066370 001101          BNE $OVER ;;LOOP ON PRESENT TEST?
                                ;;YES IF SW14=1
066372 000416          ;#####START OF CODE FOR THE XOR TESTER#####
                                $XTSTR: BR 6$ \ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
066374 013746 000004          MOV @#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
066400 012737 066420 000004          MOV #5,@#ERRVEC ;;SET FOR TIMEOUT
066406 005737 177060          TST @#177060 ;;TIME OUT ON XOR?
066412 012637 000004          MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
066416 000453          BR $SVLAD ;;GO TO THE NEXT TEST
066420 022626          5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
066422 012637 000004          MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
066426 000413          BR 7$ ;;LOOP ON THE PRESENT TEST
066430          6$:;#####END OF CODE FOR THE XOR TESTER#####
066430 105737 001103          2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
066434 001421          BEQ 3$ ;;BR IF NO
066436 123737 001115 001103          CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
066444 101015          BHI 3$ ;;BR IF NO
066446 032777 001000 112464          BIT #BIT09,@SWR ;;LOOP ON ERROR?
066454 001404          BEQ 4$ ;;BR IF NO
066456 013737 001110 001106          7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
066464 000443          BR $OVER
066466 105037 001103          4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
066472 005037 001176          CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
066476 000415          BR 1$ ;;ESCAPE TO THE NEXT TEST
066500 032777 004000 112432          3$: BIT #BIT11,@SWR ;;INHIBIT ITERATIONS?
066506 001011          BNE 1$ ;;BR IF YES
066510 005737 001100          TST $PASS ;;IF FIRST PASS OF PROGRAM
066514 001406          BEQ 1$ ;; INHIBIT ITERATIONS
066516 005237 001104          INC $ICNT ;;INCREMENT ITERATION COUNT
066522 023737 001176 001104          CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
066530 002021          BGE $OVER ;;BR IF MORE ITERATION REQUIRED
066532 012737 000001 001104          1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
066540 013737 066610 001176          MO' $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
066546 105237 001102          $SVLAD: INCB $STNM ;;COUNT TEST NUMBERS
066552 011637 001106          MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
066556 011637 001110          MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
066562 005037 001200          CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
066566 112737 000001 001115          MOVB #1,$ERMAX ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
066574 013777 001102 112340          $OVER: MOV $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
066602 013716 001106          MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
066606 000002          RTI ;;FIXES PS
    
```



5 066610 000004

SMXCNT: 4. ;:MAX. NUMBER OF ITERATIONS  
.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 $ERRTYP 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
;*CALL
;:
;: * ERROR N ;:ERROR=EMT AND N=ERROR ITEM NUMBER
    
```

066612  
066612 104407  
066614 113737 001102 001246  
066622 105237 001103  
066626 001775  
066630 013777 001102 112304  
066636 032777 002000 112274  
066644 001402  
066646 104401 001202  
066652 005237 001112  
066656 011637 001116  
066662 162737 000002 001116  
066670 117737 112222 001114  
066676 032777 020000 112234  
066704 001004  
066706 004737 066744  
066712 104401 001207  
066716  
066716 005777 112216  
066722 100002  
066724 000000  
066726 104407  
066730  
066730 022737 066104 000042  
066736 001001  
066740 000000  
066742  
066742 000002

```

$ERROR:
    CKSWR ;:TEST FOR CHANGE IN SOFT-SWR
    MOVB $TSTNM,TSTNUM
    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
    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, $ERRTYP ;:GO TO USER ERROR ROUTINE
    TYPE , $CRLF

    20$:
    2$: TST @SWR ;:HALT ON ERROR
        BPL 3$ ;:SKIP IF CONTINUE
        HALT ;:HALT ON ERROR!
        CKSWR ;:TEST FOR CHANGE IN SOFT-SWR

    3$:
    CMP # $SENDAD, @ #42 ;:ACT-11 AUTO-ACCEPT?
    BNE 6$ ;:BRANCH IF NO
    HALT ;:YES

    6$:
    RTI ;:RETURN
    
```

6

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

;:*****
;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
    
```

066744  
066744 104401 001207  
066750 010046  
066752 005000  
066754 153700 001114  
066760 001004  
  
066762 013746 001116

```

$ERRTYP:
    TYPE , $CRLF ;: "CARRIAGE RETURN" & "LINE FEED"
    MOV RO, -(SP) ;: SAVE RO
    CLR RO ;: PICKUP THE ITEM INDEX
    BISB @ # $ITEMB, RO
    BNE 1$ ;: IF ITEM NUMBER IS ZERO, JUST
    ;: TYPE THE PC OF THE ERROR
    MOV $ERRPC, -(SP) ;: SAVE $ERRPC FOR TYPEOUT
    
```



```

066766 104402
066770 000445
066772 005300
066774 006300
066776 006300
067000 006300
067002 062700 001310
067006 012037 067016
067012 001404
067014 104401
067016 000000
067020 104401 001207
067024 012037 067034
067030 001404
067032 104401
067034 000000
067036 104401 001207
067042 010146
067044 012001
067046 001415
067050 012000
067052 105720
067054 001003
067056 013146
067060 104402
067062 000402
067064
067064 013146
067066 104405
067070 005711
067072 001403
067074 104401 067114
067100 000764

067102 012601
067104 012600
067106 104401 001207
067112 000207
067114 040 040 000

TYPOC
BR 10$
1$: DEC RO
ASL RO
ASL RO
ASL RO
ADD #ERRTB,RO
MOV (RO)+,2$
BEQ 3$
TYPE
2$: .WORD 0
TYPE ,SCLF
3$: MOV (RO)+,4$
BEQ 5$
TYPE
4$: .WORD 0
TYPE ,SCLF
5$: MOV R1,-(SP)
MOV (RO)+,R1
BEQ 9$
MOV (RO)+,RO
6$: TSTB (RO)+
BNE 7$
MOV @ (R1)+,-(SP)
TYPOC
BR 8$
7$: MOV @ (R1)+,-(SP)
TYPDS
8$: TST (R1)
BEQ 9$
TYPE ,11$
BR 6$
9$: MOV (SP)+,R1
10$: MOV (SP)+,RO
TYPE ,SCLF
RTS PC
11$: .ASCIZ / /
.EVEN
.SBTTL TYPE ROUTINE

```

```

:::ERROR ADDRESS
:::GO TYPE--OCTAL ASCII(ALL DIGITS)
:::GET OUT
:::ADJUST THE INDEX SO THAT IT WILL
::: WORK FOR THE ERROR TABLE

:::FORM TABLE POINTER
:::PICKUP "ERROR MESSAGE" POINTER
:::SKIP TYPEOUT IF NO POINTER
:::TYPE THE "ERROR MESSAGE"
:::"ERROR MESSAGE" POINTER GOES HERE
:::"CARRIAGE RETURN" & "LINE FEED"
:::PICKUP "DATA HEADER" POINTER
:::SKIP TYPEOUT IF 0
:::TYPE THE "DATA HEADER"
:::"DATA HEADER" POINTER GOES HERE
:::"CARRIAGE RETURN" & "LINE FEED"
:::SAVE R1
:::PICKUP "DATA TABLE" POINTER
:::BR IF NO DATA TO BE TYPED
:::PICKUP "DATA FORMAT" POINTER
:::"OCTAL" OR "DECIMAL"
:::BR IF DECIMAL
:::SAVE @ (R1)+ FOR TYPEOUT
:::GO TYPE--OCTAL ASCII(ALL DIGITS)

:::SAVE @ (R1)+ FOR TYPEOUT
:::GO TYPE--DECIMAL ASCII WITH SIGN
:::IS THERE ANOTHER NUMBER?
:::BR IF NO
:::TYPE TWO(2) SPACES
:::LOOP

:::RESTORE R1
:::RESTORE RO
:::"CARRIAGE RETURN" & "LINE FEED"
:::RETURN
:::TWO(2) SPACES

```

```

*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
*
*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*

```

```

067120 105737 001157 $TYPE: TSTB $STPFLG ::IS THERE A TERMINAL?
067124 100002 BPL 1$ ::BR IF YES
067126 000000 HALT ::HALT HERE IF NO TERMINAL
067130 000407 BR 3$ ::LEAVE
067132 010046 1$: MOV RO,-(SP) ::SAVE RO
067134 017600 000002 MOV @2(SP),RO ::GET ADDRESS OF ASCIZ STRING
067140 112046 2$: MOV (RO)+,-(SP) ::PUSH CHARACTER TO BE TYPED ONTO STACK
067142 001005 BNE 4$ ::BR IF IT ISN'T THE TERMINATOR
067144 005726 TST (SP)+ ::IF TERMINATOR POP IT OFF THE STACK
067146 012600 60$: MOV (SP)+,RO ::RESTORE RO
067150 062716 000002 3$: ADD #2,(SP) ::ADJUST RETURN PC
067154 000002 RTI ::RETURN
067156 122716 000011 4$: CMPB #HT,(SP) ::BRANCH IF <HT>
067162 001430 BEQ 8$
067164 122716 000200 CMPB #CRLF,(SP) ::BRANCH IF NOT <CRLF>
067170 001006 BNE 5$
067172 005726 TST (SP)+ ::POP <CR><LF> EQUIV
067174 104401 TYPE ::TYPE A CR AND LF
067176 001207 $CRLF
067200 105037 067406 CLRB $CHARCNT ::CLEAR CHARACTER COUNT
067204 000755 BR 2$ ::GET NEXT CHARACTER
067206 004737 067270 5$: JSR PC,$TYPEC ::GO TYPE THIS CHARACTER
067212 123726 001156 6$: CMPB $FILLC,(SP)+ ::IS IT TIME FOR FILLER CHARS.?
067216 001350 BNE 2$ ::IF NO GO GET NEXT CHAR.
067220 013746 001154 MOV $NULL,-(SP) ::GET # OF FILLER CHARS. NEEDED
::AND THE NULL CHAR.
067224 105366 000001 7$: DECB 1(SP) ::DOES A NULL NEED TO BE TYPED?
067230 002770 BLT 6$ ::BR IF NO--GO POP THE NULL OFF OF STACK
067232 004737 067270 JSR PC,$TYPEC ::GO TYPE A NULL
067236 105337 067406 DECB $CHARCNT ::DO NOT COUNT AS A COUNT
067242 000770 BR 7$ ::LOOP

```

;HORIZONTAL TAB PROCESSOR

```

067244 112716 000040 8$: MOVB #' ,(SP) ::REPLACE TAB WITH SPACE
067250 004737 067270 9$: JSR PC,$TYPEC ::TYPE A SPACE
067254 132737 000007 067406 BITB #7,$CHARCNT ::BRANCH IF NOT AT
067262 001372 BNE 9$ ::TAB STOP
067264 005726 TST (SP)+ ::POP SPACE OFF STACK
067266 000724 BR 2$ ::GET NEXT CHARACTER
067270 $TYPEC:
067270 105777 111650 TSTB @$TKS ::CHAR IN KYBD BUFFER?
067274 100022 BPL 10$ ::BR IF NOT
067276 017746 111644 MOV @$TKB,-(SP) ::GET CHAR
067302 042716 177600 BIC #177600,(SP) ::STRIP EXTRANEIOUS BITS
067306 122716 000023 CMPB #$XOFF,(SP) ::WAS CHAR XOFF
067312 001012 BNE 102$ ::BR IF NOT
067314 105777 111624 101$: TSTB @$TKS ::WAIT FOR CHAR
067320 100375 BPL 101$
067322 117716 111620 MOVB @$TKB,(SP) ::GET CHAR
067326 042716 177600 BIC #177600,(SP) ::STRIP IT
067332 122716 000021 CMPB #$XON,(SP) ::WAS IT XON?
067336 001366 BNE 101$ ::BR IF NOT
067340 005726 102$: TST (SP)+ ::FIX STACK

```



```

067342      10$:
067342 105777 111602      TSTB   @ $TPS      ;;WAIT UNTIL PRINTER IS READY
067346 100375      BPL     10$
067350 116677 000002 111574  MOVB   2(SP),@ $TPB  ;;LOAD CHAR TO BE TYPED INTO DATA REG.
067356 122766 000015 000002  CMPB   #CR,2(SP)    ;;IS CHARACTER A CARRIAGE RETURN?
067364 001003      BNE     1$          ;;BRANCH IF NO
067366 105037 067406      CLRB   $CHARCNT    ;;YES--CLEAR CHARACTER COUNT
067372 000406      BR      $TYPEX     ;;EXIT
067374 122766 000012 000002 1$:  CMPB   #LF,2(SP)    ;;IS CHARACTER A LINE FEED?
067402 001402      BEQ     $TYPEX     ;;BRANCH IF YES
067404 105227      INCB   (PC)+      ;;COUNT THE CHARACTER
067406 000000      $CHARCNT:.WORD 0  ;;CHARACTER COUNT STORAGE
067410 000207      $TYPEX: RTS     PC
    
```

8

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOS   ;;CALL FOR TYPEOUT
*   .BYTE  N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*   .BYTE  M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPON   ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*   MOV     NUM,-(SP)      ;;NUMBER TO BE TYPED
*   TYPOC   ;;CALL FOR TYPEOUT
    
```

```

067412 017646 000000      $TYPOS: MOV   @ (SP),-(SP)  ;;PICKUP THE MODE
067416 116637 000001 067635  MOVB   1(SP),$OFILL  ;;LOAD ZERO FILL SWITCH
067424 112637 067637      MOVB   (SP)+,$OMODE+1 ;;NUMBER OF DIGITS TO TYPE
067430 062716 000002      ADD    #2,(SP)      ;;ADJUST RETURN ADDRESS
067434 000406      BR     $TYPON
067436 112737 000001 067635  $TYPOC: MOVB  #1,$OFILL  ;;SET THE ZERO FILL SWITCH
067444 112737 000006 067637  MOVB   #6,$OMODE+1  ;;SET FOR SIX(6) DIGITS
067452 112737 000005 067634  $TYPON: MOVB  #5,$OCNT  ;;SET THE ITERATION COUNT
067460 010346      MOV    R3,-(SP)    ;;SAVE R3
067462 010446      MOV    R4,-(SP)    ;;SAVE R4
067464 010546      MOV    R5,-(SP)    ;;SAVE R5
067466 113704 067637      MOVB   $OMODE+1,R4  ;;GET THE NUMBER OF DIGITS TO TYPE
067472 005404      NEG    R4
067474 062704 000006      ADD    #6,R4        ;;SUBTRACT IT FOR MAX. ALLOWED
067500 110437 067636      MOVB   R4,$OMODE   ;;SAVE IT FOR USE
067504 113704 067635      MOVB   $OFILL,R4   ;;GET THE ZERO FILL SWITCH
067510 016605 000012      MOV    12(SP),R5   ;;PICKUP THE INPUT NUMBER
067514 005003      CLR    R3          ;;CLEAR THE OUTPUT WORD
    
```

```

067516 006105      1$:  ROL    R5      ;;ROTATE MSB INTO 'C'
067520 000404      BR     3$      ;;GO DO MSB
067522 006105      2$:  ROL    R5      ;;FORM THIS DIGIT
067524 006105      ROL    R5
067526 006105      ROL    R5
067530 010503      MOV    R5,R3
067532 006103      3$:  ROL    R3      ;;GET LSB OF THIS DIGIT
067534 105337 067636 DECB   $OMODE   ;;TYPE THIS DIGIT?
067540 100016      BPL   7$      ;;BR IF NO
067542 042703 177770 BIC   #177770,R3 ;;GET RID OF JUNK
067546 001002      BNE   4$      ;;TEST FOR 0
067550 005704      TST   R4      ;;SUPPRESS THIS 0?
067552 001403      BEQ   5$      ;;BR IF YES
067554 005204      4$:  INC    R4      ;;DON'T SUPPRESS ANYMORE 0'S
067556 052703 000060 BIS   #'0,R3   ;;MAKE THIS DIGIT ASCII
067562 052703 000040 5$:  BIS   #' ,R3  ;;MAKE ASCII IF NOT ALREADY
067566 110337 067632 MOVB  R3,8$    ;;SAVE FOR TYPING
067572 104401 067632 TYPE  ,8$      ;;GO TYPE THIS DIGIT
067576 105337 067634 7$:  DECB  $OCNT   ;;COUNT BY 1
067602 003347      BGT   2$      ;;BR IF MORE TO DO
067604 002402      BLT   6$      ;;BR IF DONE
067606 005204      INC   R4      ;;INSURE LAST DIGIT ISN'T A BLANK
067610 000744      BR    2$      ;;GO DO THE LAST DIGIT
067612 012605      6$:  MOV   (SP)+,R5 ;;RESTORE R5
067614 012604      MOV   (SP)+,R4 ;;RESTORE R4
067616 012603      MOV   (SP)+,R3 ;;RESTORE R3
067620 016666 000002 000004 MOV   2(SP),4(SP) ;;SET THE STACK FOR RETURNING
067626 012616      MOV   (SP)+,(SP)
067630 000002      RTI                    ;;RETURN
067632 000      8$:  .BYTE  0      ;;STORAGE FOR ASCII DIGIT
067633 000      .BYTE  0      ;;TERMINATOR FOR TYPE ROUTINE
067634 000      $OCNT: .BYTE  0      ;;OCTAL DIGIT COUNTER
067635 000      $OFILL: .BYTE  0      ;;ZERO FILL SWITCH
067636 000000      $OMODE: .WORD  0      ;;NUMBER OF DIGITS TO TYPE
9  .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

```

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
*REPLACED WITH SPACES.

```

```

*CALL:
*   MOV    NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
*   TYPDS                    ;;GO TO THE ROUTINE

```

```

067640 010046      $TYPDS: MOV   R0,-(SP)      ;;PUSH R0 ON STACK
067642 010146      MOV   R1,-(SP)      ;;PUSH R1 ON STACK
067644 010246      MOV   R2,-(SP)      ;;PUSH R2 ON STACK
067646 010346      MOV   R3,-(SP)      ;;PUSH R3 ON STACK
067650 010546      MOV   R5,-(SP)      ;;PUSH R5 ON STACK
067652 012746 020200 MOV   #20200,-(SP)   ;;SET BLANK SWITCH AND SIGN
067656 016605 000020 MOV   20(SP),R5      ;;GET THE INPUT NUMBER
067662 100004      BPL   1$      ;;BR IF INPUT IS POS.
067664 005405      NEG   R5      ;;MAKE THE BINARY NUMBER POS.
067666 112766 000055 000001 MOVB  #'-,1(SP)      ;;MAKE THE ASCII NUMBER NEG.

```



```

067674 005000          1$: CLR R0          ;;ZERO THE CONSTANTS INDEX
067676 012703 070054  MOV #SDBLK,R3      ;;SETUP THE OUTPUT POINTER
067702 112723 000040  MOVB #' ,(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
067706 005002          2$: CLR R2          ;;CLEAR THE BCD NUMBER
067710 016001 070044  MOV $DTBL(R0),R1  ;;GET THE CONSTANT
067714 160105          3$: SUB R1,R5        ;;FORM THIS BCD DIGIT
067716 002402          BLT 4$          ;;BR IF DONE
067720 005202          INC R2          ;;INCREASE THE BCD DIGIT BY 1
067722 000774          BR 3$
067724 060105          4$: ADD R1,R5        ;;ADD BACK THE CONSTANT
067726 005702          TST R2          ;;CHECK IF BCD DIGIT=0
067730 001002          BNE 5$          ;;FALL THROUGH IF 0
067732 105716          TSTB (SP)        ;;STILL DOING LEADING 0'S?
067734 100407          BMI 7$          ;;BR IF YES
067736 106316          5$: ASLB (SP)        ;;MSD?
067740 103003          BCC 6$          ;;BR IF NO
067742 116663 000001 177777  MOVB 1(SP),-1(R3)  ;;YES--SET THE SIGN
067750 052702 000060  6$: BIS #'0,R2      ;;MAKE THE BCD DIGIT ASCII
067754 052702 000040  7$: BIS #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
067760 110223          MOVB R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
067762 005720          TST (R0)+        ;;JUST INCREMENTING
067764 020027 000010  CMP R0,#10       ;;CHECK THE TABLE INDEX
067770 002746          BLT 2$          ;;GO DO THE NEXT DIGIT
067772 003002          BGT 8$          ;;GO TO EXIT
067774 010502          MOV R5,R2        ;;GET THE LSD
067776 000764          BR 6$          ;;GO CHANGE TO ASCII
070000 105726          8$: TSTB (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
070002 100003          BPL 9$          ;;BR IF NO
070004 116663 177777 177776  MOVB -1(SP),-2(R3)  ;;YES--SET THE SIGN FOR TYPING
070012 105013          9$: CLRB (R3)        ;;SET THE TERMINATOR
070014 012605          MOV (SP)+,R5     ;;POP STACK INTO R5
070016 012603          MOV (SP)+,R3     ;;POP STACK INTO R3
070020 012602          MOV (SP)+,R2     ;;POP STACK INTO R2
070022 012601          MOV (SP)+,R1     ;;POP STACK INTO R1
070024 012600          MOV (SP)+,R0     ;;POP STACK INTO R0
070026 104401 070054  TYPE ,SDBLK        ;;NOW TYPE THE NUMBER
070032 016666 000002 000004  MOV 2(SP),4(SP)    ;;ADJUST THE STACK
070040 012616          MOV (SP)+,(SP)
070042 000002          RTI          ;;RETURN TO USER
070044 023420          $DTBL: 10000.
070046 001750          1000.
070050 000144          100.
070052 000012          10.
070054          $DBLK: .BLKW 4
          .SBTTL TTY INPUT ROUTINE

*****
.ENABL LSB
$TKCNT: .WORD 0          ;;NUMBER OF ITEMS IN QUEUE
$TKQIN: .WORD 0         ;;INPUT POINTER
$TKQOUT: .WORD 0        ;;OUTPUT POINTER
$TKQSRT: .BLKB 1        ;;TTY KEYBOARD QUEUE
$TKQEND=.
.EVEN

;*TK INITIALIZE ROUTINE
;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
    
```

10.

```

;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
;*CALL:
;*      JSR      PC,$TKINT
;*      RETURN
$TKINT: CLR      $TKCNT      ;;CLEAR COUNT OF ITEMS IN QUEUE
        MOV      #$TKQSR, $TKQIN ;;MOVE THE STARTING ADDRESS OF THE
        MOV      $TKQIN, $TKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
        MOV      #$TKSRV, @#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
        MOV      #200, @#TKVEC+2 ;;'BR' LEVEL 4
        TST      @ $TKB      ;;CLEAR DONE FLAG
        MOV      #100, @ $TKS ;;ENABLE TTY KEYBOARD INTERRUPT
        RTS      PC          ;;RETURN TO CALLER

;*TK SERVICE ROUTINE
;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
;*IT IN THE QUEUE.
$TKSRV: MOV      @ $TKB, -(SP) ;;PICKUP THE CHARACTER
        BIC      #^C177, (SP) ;;STRIP THE JUNK
1$:     CMP      (SP), #7      ;;IS IT A CONTROL G?
        BNE      2$          ;;BRANCH IF NO
        CMP      #SWREG, SWR   ;;IS SOFT-SWR SELECTED?
        BEQ      6$          ;;GO TO SWR CHANGE

2$:     CMP      #1, $TKCNT     ;;IS THE QUEUE FULL?
        BNE      3$          ;;BRANCH IF NO
        TYPE      , $BELL      ;;RING THE TTY BELL
        TST      (SP)+        ;;CLEAN CHARACTER OFF OF STACK
        BR       5$          ;;EXIT
3$:     CMP      (SP), #23     ;;IS IT A CONTROL-S?
        BNE      32$         ;;BRANCH IF NO
        CLR      @ $TKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
        TST      (SP)+        ;;CLEAN CHAR OFF STACK
31$:    TST      @ $TKS      ;;WAIT FOR A CHAR
        BPL      31$         ;;LOOP UNTIL ITS THERE
        MOV      @ $TKB, -(SP) ;;GET THE CHARACTER
        BIC      #^C177, (SP) ;;MAKE IT 7-BIT ASCII
        CMP      (SP)+, #21    ;;IS IT A CONTROL-Q?
        BNE      31$         ;;BRANCH IF NO
        MOV      #100, @ $TKS ;;REENABLE TTY KEYBOARD INTERRUPTS
        RTI      ;;RETURN
32$:    INC      $TKCNT      ;;COUNT THIS CHARACTER
        CMP      (SP), #140    ;;IS IT UPPER CASE?
        BLT      4$          ;;BRANCH IF YES
        CMP      (SP), #175    ;;IS IT A SPECIAL CHAR?
        BGT      4$          ;;BRANCH IF YES
        BIC      #40, (SP)    ;;MAKE IT UPPER CASE
4$:     MOV      (SP)+, @ $TKQIN ;;AND PUT IT IN QUEUE
        INC      $TKQIN      ;;UPDATE THE POINTER
        CMP      $TKQIN, # $TKQEND ;;GO OFF THE END?
        BNE      5$          ;;BRANCH IF NO
        MOV      #$TKQSR, $TKQIN ;;RESET THE POINTER
5$:     RTI      ;;RETURN

```



```

*****
*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
070336 022737 000176 001140 $CKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED
070344 001104 BNE 15$ ;;EXIT IF NOT
070346 105777 110572 TSTB @TKS ;;IS A CHAR WAITING?
070352 100101 BPL 15$ ;;IF NOT, EXIT
070354 117746 110566 MOVB @TKB,-(SP) ;;YES
070360 042716 177600 BIC #^C177,(SP) ;;MAKE IT 7-BIT ASCII
070364 021627 000007 CMP (SP),#7 ;;IS IT A CONTROL-G?
070370 001300 BNE 2$ ;;IF NOT, PUT IT IN THE TTY QUEUE
;;AND EXIT

*****
*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
070372 123727 001134 000001 6$: CMPB $AUTOB,#1 ;;ARE WE RUNNING IN AUTO-MODE?
070400 001674 BEQ 2$ ;;BRANCH IF YES
070402 005726 TST (SP)+ ;;CLEAR CONTROL-G OFF STACK
070404 004737 070074 JSR PC,$TKINT ;;FLUSH THE TTY INPUT QUEUE
070410 005077 110530 CLR @TKS ;;DISABLE TTY KEYBOARD INTERRUPTS
070414 112737 000001 001135 MOVB #1,$INTAG ;;SET INTERRUPT MODE INDICATOR

070422 104401 071200 TYPE ,SCNTLG ;;ECHO THE CONTROL-G (^G)
070426 104401 071205 $GTSWR: TYPE ,SMSWR ;;TYPE CURRENT CONTENTS
070432 013746 000176 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT
070436 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
070440 104401 071216 TYPE ,SMNEW ;;PROMPT FOR NEW SWR
070444 005046 19$: CLR -(SP) ;;CLEAR COUNTER
070446 005046 CLR -(SP) ;;THE NEW SWR
070450 105777 110470 7$: TSTB @TKS ;;CHAR THERE?
070454 100375 BPL 7$ ;;IF NOT TRY AGAIN

070456 117746 110464 MOVB @TKB,-(SP) ;;PICK UP CHAR
070462 042716 177600 BIC #^C177,(SP) ;;MAKE IT 7-BIT ASCII

070466 021627 000025 9$: CMP (SP),#25 ;;IS IT A CONTROL-U?
070472 001005 BNE 10$ ;;BRANCH IF NOT
070474 104401 071173 TYPE ,SCNTLU ;;YES, ECHO CONTROL-U (^U)
070500 062706 000006 20$: ADD #6,SP ;;IGNORE PREVIOUS INPUT
070504 000757 BR 19$ ;;LET'S TRY IT AGAIN

070506 021627 000015 10$: CMP (SP),#15 ;;IS IT A <CR>?
070512 001022 BNE 16$ ;;BRANCH IF NO
070514 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
070520 001403 BEQ 11$ ;;BRANCH IF YES
070522 016677 000002 110410 MOV 2(SP),@SWR ;;SAVE NEW SWR
070530 062706 000006 11$: ADD #6,SP ;;CLEAR UP STACK
070534 104401 001207 14$: TYPE ,SCRLF ;;ECHO <CR> AND <LF>
070540 123727 001135 000001 CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?

```

```
070546 001003          BNE      15$          ;;BRANCH IF NOT
070550 012777 000100 110366  MOV     #100,@$TKS    ;;RE-ENABLE TTY KBD INTERRUPTS
070556 000002          RTI          ;;RETURN
070560 004737 067270 15$: JSR     PC,$TYPEC    ;;ECHO CHAR
070564 021627 000060 16$: CMP     (SP),#60    ;;CHAR < 0?
070570 002420          BLT     18$          ;;BRANCH IF YES
070572 021627 000067  CMP     (SP),#67    ;;CHAR > 7?
070576 003015          BGT     18$          ;;BRANCH IF YES
070600 042726 000060  BIC     #60,(SP)+   ;;STRIP-OFF ASCII
070604 005766 000002  TST     2(SP)       ;;IS THIS THE FIRST CHAR
070610 001403          BEQ     17$          ;;BRANCH IF YES
070612 006316          ASL     (SP)        ;;NO, SHIFT PRESENT
070614 006316          ASL     (SP)        ;; CHAR OVER TO MAKE
070616 006316          ASL     (SP)        ;; ROOM FOR NEW ONE.
070620 005266 000002 17$: INC     2(SP)       ;;KEEP COUNT OF CHAR
070624 056616 177776  BIS     -2(SP),(SP) ;;SET IN NEW CHAR
070630 000707          BR      7$          ;;GET THE NEXT ONE
070632 104401 001206 18$: TYPE  $QUES      ;;TYPE ?<CR><LF>
070636 000720          BR      20$         ;;SIMULATE CONTROL-U
.DSABL  LSB
```

```
*****
*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
*CALL:
*      RDCHR          ;;GET A CHARACTER FROM THE QUEUE
*      RETURN HERE   ;;CHARACTER IS ON THE STACK
*                   ;;WITH PARITY BIT STRIPPED OFF
*
```

```
070640 011646          $RDCHR: MOV     (SP),-(SP)  ;;PUSH DOWN THE PC AND
070642 016666 000004 000002  MOV     4(SP),2(SP)  ;;THE PS
070650 005066 000004          CLR     4(SP)       ;;GET READY FOR A CHARACTER
070654 005046          CLR     -(SP)      ;;PUT NEW PS ON STACK
070656 012746 070664          MOV     #64$,-(SP) ;;PUT NEW PC ON STACK
070662 000002          RTI          ;;POP NEW PC AND PS
070664          64$:
070664 005737 070064 1$: TST     $TKCNT     ;;WAIT ON A CHARACTER
070670 001775          BEQ     1$          ;;
070672 005337 070064          DEC     $TKCNT     ;;DECREMENT THE COUNTER
070676 117766 177166 000004  MOVB    @$TKQOUT,4(SP) ;;GET ONE CHARACTER
070704 005237 070070          INC     $TKQOUT   ;;UPDATE THE POINTER
070710 023727 070070 070073  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
070716 001003          BNE     2$          ;;BRANCH IF NO
070720 012737 070072 070070  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
070726 000002          RTI          ;;RETURN
2$:
```

```
*****
*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
*
```

```
070730 010346          $RDLIN: MOV     R3,-(SP)  ;;SAVE R3
070732 005046          CLR     -(SP)      ;;CLEAR THE RUBOUT KEY
070734 012703 071164 1$: MOV     #$TTYIN,R3  ;;GET ADDRESS
070740 022703 071173 2$: CMP     #$TTYIN+7,R3 ;;BUFFER FULL?
```



```

070744 101456          BLOS      4$          ;;BR IF YES
070746 104410          RDCHR                    ;;GO READ ONE CHARACTER FROM THE TTY
070750 112613          MOVVB     (SP)+,(R3)      ;;GET CHARACTER
070752 122713 000177 10$:  CMPB     #177,(R3)      ;;IS IT A RUBOUT
070756 001022          BNE      5$          ;;BR IF NO
070760 005716          TST      (SP)          ;;IS THIS THE FIRST RUBOUT?
070762 001007          BNE      6$          ;;BR IF NO
070764 112737 000134 071162 MOVVB     #'\,9$        ;;TYPE A BACK SLASH
070772 104401 071162          TYPE     ,9$
070776 012716 177777          MOV      #-1,(SP)      ;;SET THE RUBOUT KEY
071002 005303          DEC      R3          ;;BACKUP BY ONE
071004 020327 071164          CMP      R3,#$TTYIN   ;;STACK EMPTY?
071010 103434          BLO      4$          ;;BR IF YES
071012 111337 071162          MOVVB     (R3),9$      ;;SETUP TO TYPEOUT THE DELETED CHAR.
071016 104401 071162          TYPE     ,9$        ;;GO TYPE
071022 000746          BR       2$          ;;GO READ ANOTHER CHAR.
071024 005716          TST      (SP)          ;;RUBOUT KEY SET?
071026 001406          BEQ      7$          ;;BR IF NO
071030 112737 000134 071162 MOVVB     #'\,9$        ;;TYPE A BACK SLASH
071036 104401 071162          TYPE     ,9$
071042 005016          CLR      (SP)          ;;CLEAR THE RUBOUT KEY
071044 122713 000025 7$:  CMPB     #25,(R3)      ;;IS CHARACTER A CTRL U?
071050 001003          BNE      8$          ;;BR IF NO
071052 104401 071173          TYPE     , $CNTLU    ;;TYPE A CONTROL 'U'
071056 000726          BR       1$          ;;GO START OVER
071060 122713 000022 8$:  CMPB     #22,(R3)      ;;IS CHARACTER A '^R'?
071064 001011          BNE      3$          ;;BRANCH IF NO
071066 105013          CLRB    (R3)          ;;CLEAR THE CHARACTER
071070 104401 001207          TYPE     , $CRLF    ;;TYPE A 'CR' & 'LF'
071074 104401 071164          TYPE     , $TTYIN   ;;TYPE THE INPUT STRING
071100 000717          BR       2$          ;;GO PICKUP ANOTHER CHACTER
071102 104401 001206 4$:  TYPE     , $QUES     ;;TYPE A '?'
071106 000712          BR       1$          ;;CLEAR THE BUFFER AND LOOP
071110 111337 071162 3$:  MOVVB     (R3),9$      ;;ECHO THE CHARACTER
071114 104401 071162          TYPE     ,9$
071120 122723 000015          .CMPB    #15,(R3)+   ;;CHECK FOR RETURN
071124 001305          BNE      2$          ;;LOOP IF NOT RETURN
071126 105063 177777          CLRB    -1(R3)      ;;CLEAR RETURN (THE 15)
071132 104401 001210          TYPE     , $LF      ;;TYPE A LINE FEED
071136 005726          TST      (SP)+      ;;CLEAN RUBOUT KEY FROM THE STACK
071140 012603          MOV      (SP)+,R3    ;;RESTORE R3
071142 011646          MOV      (SP),-(SP)  ;;ADJUST THE STACK AND PUT ADDRESS OF THE
071144 016666 000004 000002 MOV      4(SP),2(SP)  ;; FIRST ASCII CHARACTER ON IT
071152 012766 071164 000004 MOV      #$TTYIN,4(SP)
071160 000002          RTI                    ;;RETURN
071162          000          9$:  .BYTE    0          ;;STORAGE FOR ASCII CHAR. TO TYPE
071163          000          .BYTE    0          ;;TERMINATOR
071164          .BLKB   7          ;;RESERVE 7 BYTES FOR TTY INPUT
071173          136          125          015  $TTYIN: .ASCIZ  /^U/<15><12>  ;;CONTROL 'U'
071200          136          107          015  $CNTLG: .ASCIZ  /^G/<15><12>  ;;CONTROL 'G'
071205          015          012          123  $MSWR: .ASCIZ  <15><12>/SWR = /
071216          040          040          116  $MNEW: .ASCIZ  / NEW = /
          .EVEN
          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

```

\*\*\*\*\*  
\*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND

```

;*CHANGE IT TO BINARY.
;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
;*CALL:
;*      RDOCT          ;;READ AN OCTAL NUMBER
;*      RETURN HERE   ;;LOW ORDER BITS ARE ON TOP OF THE STACK
;*                  ;;HIGH ORDER BITS ARE IN $HIOCT
    
```

```

071230 011646
071232 016666 000004 000002
071240 010046
071242 010146
071244 010246
071246 104411
071250 012600
071252 010037 071356
071256 005001
071260 005002
071262 112046
071264 001420
071266 122716 000060
071272 003026
071274 122716 000067
071300 002423
071302 006301
071304 006102
071306 006301
071310 006102
071312 006301
071314 006102
071316 042716 177770
071322 062601
071324 000756
071326 005726
071330 010166 000012
071334 010237 071366
071340 012602
071342 012601
071344 012600
071346 000002
071350 005726
071352 105010
071354 104401
071356 000000
071360 104401 001206
071364 000730
071366 000000
    
```

```

$RDOCT: MOV      (SP),-(SP)      ;;PROVIDE SPACE FOR THE
        MOV      4(SP),2(SP)    ;;INPUT NUMBER
        MOV      R0,-(SP)       ;;PUSH R0 ON STACK
        MOV      R1,-(SP)       ;;PUSH R1 ON STACK
        MOV      R2,-(SP)       ;;PUSH R2 ON STACK
1$:     RDLIN                      ;;READ AN ASCII LINE
        MOV      (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
        MOV      R0,5$          ;;AND SAVE IT
        CLR      R1              ;;CLEAR DATA WORD
        CLR      R2
2$:     MOVB     (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
        BEQ      3$              ;;IF ZERO GET OUT
        CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
        BGT      4$              ;;IS AN OCTAL DIGIT
        CMPB    #'7,(SP)
        BLT      4$
        ASL     R1                ;;*2
        ROL     R2
        ASL     R1                ;;*4
        ROL     R2
        ASL     R1                ;;*8
        ROL     R2
        BIC     #'C7,(SP)        ;;STRIP THE ASCII JUNK
        ADD     (SP)+,R1         ;;ADD IN THIS DIGIT
        BR      2$              ;;LOOP
3$:     TST     (SP)+            ;;CLEAN TERMINATOR FROM STACK
        MOV     R1,12(SP)        ;;SAVE THE RESULT
        MOV     R2,$HIOCT
        MOV     (SP)+,R2        ;;POP STACK INTO R2
        MOV     (SP)+,R1        ;;POP STACK INTO R1
        MOV     (SP)+,R0        ;;POP STACK INTO R0
        RTI                      ;;RETURN
4$:     TST     (SP)+            ;;CLEAN PARTIAL FROM STACK
        CLRB   (R0)              ;;SET A TERMINATOR
        TYPE   0                 ;;TYPE UP THRU THE BAD CHAR.
5$:     .WORD   0
        TYPE   $QUES             ;;'"?' 'CR' & 'LF'
        BR     1$                ;;TRY AGAIN
$HIOCT: .WORD   0                ;;HIGH ORDER BITS GO HERE
.SBITL  SAVE AND RESTORE R0-R5 ROUTINES
    
```

```

;*****
;*SAVE R0-R5
;*CALL:
;*      SAVREG
;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
;*
    
```



```

; *TOP---(+16)
; * +2---(+18)
; * +4---R5
; * +6---R4
; * +8---R3
; *+10---R2
; *+12---R1
; *+14---R0
    
```

```

071370
071370 010046
071372 010146
071374 010246
071376 010346
071400 010446
071402 010546
071404 016646 000022
071410 016646 000022
071414 016646 000022
071420 016646 000022
071424 000002
    
```

```

$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
    
```

```

071426
071426 012666 000022
071432 012666 000022
071436 012666 000022
071442 012666 000022
071446 012605
071450 012604
071452 012603
071454 012602
071456 012601
071460 012600
071462 000002
    
```

```

; *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
    
```

13

.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.
    
```

```

071464 010046
071466 016600 000002
071472 005740
071474 111000
071476 006300
071500 016000 071520
071504 000200
    
```

```

$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

```

071506 011646
    
```

```

$TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
    
```

```
071510 016666 000004 000002      MOV      4(SP),2(SP)      ;;MOVE THE PSW DOWN
071516 000002                      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
          $TYPE  ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
          $TYPOC ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
          $TYPOS ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
          $TYPON ;;CALL=TYPON     TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
          $TYPDS ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

071520 071506
071522 067120
071524 067436
071526 067412
071530 067452
071532 067640

071534 070426          $GTSWR  ;;CALL=GTSWR     TRAP+6(104406)  GET SOFT-SWR SETTING

071536 070336          $CKSWR  ;;CALL=CKSWR     TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
071540 070640          $RDCHR  ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
071542 070730          $RDLIN  ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
071544 071230          $RDOCT  ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
071546 071370          $SAVREG ;;CALL=SAVREG     TRAP+13(104413) SAVE R0-R5 ROUTINE
071550 071426          $RESREG ;;CALL=RESREG     TRAP+14(104414) RESTORE R0-R5 ROUTINE

```



```
1  
2  
3 071552 200 105 116 ENTERA: .ASCIZ <CRLF>/ENTER DRIVE ADDRESS: /  
4 071601 040 111 116 ADRERR: .ASCIZ / INVALID ADDRESS/<CRLF>  
5 071623 200 120 117 PORTAIS: .ASCIZ <CRLF>/PORT A ADDRESS IS: /  
6 071650 200 120 117 PORTBIS: .ASCIZ <CRLF>/PORT B ADDRESS IS: /  
7 071675 200 116 117 NOCLOCK: .ASCIZ <CRLF>/NO SYSTEM 'L' OR 'P' CLOCK/<CRLF><LF>  
8 071733 012 105 116 TESTNO: .ASCIZ <LF>/ENTER TEST #: /  
9 071753 040 111 116 BADNO: .ASCIZ / INVALID TEST NUMBER/<CRLF>  
10 072001 040 105 122 TSTERR: .ASCIZ / ERRORS/<CRLF>  
11 072012 200 012 122 ADDRIS: .ASCIZ <CRLF><LF>@RH/RM ADDRESS (RMCS1) IS: @  
12 072047 012 105 116 NTRH11: .ASCIZ <LF>@ENTER RH/RM ADDRESS: @  
13
```



Line	Code	Port	Port	Port	Code	Description
1					.SBTTL	TEST ERROR MESSAGES
2						
3	072076	127	122	117	EM1:	.ASCIZ /WRONG DRIVE TYPE/
4	072117	104	122	111	EM2:	.ASCIZ /DRIVE NOT ON LINE/
5	072141	123	105	122	EM3:	.ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
6	072223	104	122	111	EM4:	.ASCIZ /DRIVE NOT SEIZED BY PORT/
7	072254	127	122	117	EM5:	.ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
8	072322	122	105	107	EM6:	.ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
9	072422	122	105	107	EM7:	.ASCIZ /REGISTER CONTENTS WRONG AFTER RELEASE OR TIMEOUT/
10	072503	122	105	107	EM10:	.ASCIZ /REGISTER CONTENTS WRONG/
11	072533	103	117	116	EM11:	.ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
12	072617	104	122	111	EM12:	.ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
13	072667	122	105	101	EM13:	.ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
14	072754	126	117	114	EM14:	.ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
15	073017	101	124	124	EM15:	.ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
16	073076	101	124	124	EM16:	.ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
17	073151	101	124	124	EM17:	.ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
18	073230	104	122	111	EM20:	.ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
19	073310	104	122	111	EM21:	.ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
20	073363	104	122	111	EM22:	.ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
21	073450	124	111	115	EM23:	.ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
22	073516	122	105	114	EM24:	.ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
23	073575	124	111	115	EM25:	.ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
24	073640	104	122	111	EM26:	.ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
25	073725	122	105	107	EM27:	.ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
26	074003	104	122	111	EM30:	.ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
27	074100	104	122	111	EM31:	.ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
28	074155	101	124	124	EM32:	.ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
29	074226	104	122	111	EM33:	.ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
30	074330	104	122	111	EM34:	.ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
31	074433	124	111	115	EM35:	.ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
32	074512	124	111	115	EM36:	.ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
33	074564	104	122	111	EM37:	.ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
34	074632	101	124	124	EM40:	.ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
35	074707	124	111	115	EM41:	.ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
36	074751	104	122	111	EM42:	.ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
37	075037	104	122	111	EM43:	.ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
38	075114	127	122	111	EM44:	.ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
39	075171	103	117	116	EM45:	.ASCIZ @CONTROLLER SELECT SWITCH ON DRIVE NOT IN 'A/B'@
40	075250	103	101	116	EM46:	.ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
41	075317	101	124	124	EM47:	.ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
42	075405	101	124	124	EM50:	.ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR/
43	075467	101	124	124	EM51:	.ASCIZ /ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
44	075556	124	110	105	EM52:	.ASCIZ /THE ATTN BIT SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
45	075651	103	101	116	EM53:	.ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
46	075732	122	105	114	EM54:	.ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
47	076025	124	111	115	EM55:	.ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
48	076072	122	110	057	EM56:	.ASCIZ @RH/RM DIDN'T RESPOND TO ADDRESSING@
49	076135	120	117	122	EM57:	.ASCIZ /PORT REQUEST FLOP(S) WRONG STATE/
50	076176	101	124	124	EM60:	.ASCIZ /ATTENTION NOT RESET BY WRITING RMAS/
51	076242	101	124	124	EM61:	.ASCIZ /ATTENTION NOT RESET BY GO/
52	076274	101	124	124	EM62:	.ASCIZ /ATTENTION RESET BY GO WHEN NOT SEIZED/
53	076342	104	122	111	EM63:	.ASCIZ /DRIVE SEIZED BY UNIT READY CHANGE/
54	076404	101	124	124	EM64:	.ASCIZ /ATTENTION NOT SET BY UNIT READY CHANGE/
55	076453	126	117	114	EM65:	.ASCIZ /VOLUME VALID NOT RESET BY UNIT READY/
56						



1	076520	124	105	123	DH1:	.ASCIZ	/TEST #	ERR PC	PORT #	REG ADR	CONTENTS/	
2	076571	124	105	123	DH3:	.ASCIZ	/TEST #	ERR PC	REG ADR	PORT A	PORT B/	
3	076640	040	040	040	DH4:	.ASCII	/		SEIZE	ERROR/<CR><LF>		
4	076677	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #	REG ADR	GOOD BAD/
5	076763	124	105	123	DH5:	.ASCIZ	/TEST #	ERR PC	PORT #	REG ADR	GOOD	BAD/
6	077037	040	040	040	DH7:	.ASCII	/		RELSNG	ERROR/<CR><LF>		
7	077076	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #	REG ADR	GOOD BAD/
8	077162	124	105	123	DH11:	.ASCIZ	/TEST #	ERR PC	PORT #	REG ADR	CONTENTS/	
9	077233	040	040	040	DH13:	.ASCII	/		SEIZE	ERROR/<CR><LF>		
10	077272	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #	REG ADR	CONTENTS/
11	077353	040	040	040	DH22:	.ASCII	/		RELSNG	SEIZE/<CR><LF>		
12	077412	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #/		
13	077451	040	040	040	DH23:	.ASCII	/		SEIZE/<CR><LF>			
14	077500	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	REG ADR	CONTENTS/	
15	077551	040	040	040	DH26:	.ASCII	/		RELSNG/<CR><LF>			
16	077601	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #/			
17	077630	040	040	040	DH31:	.ASCII	/		RELSNG	RQSTNG/<CR><LF>		
18	077670	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #/		
19	077727	124	105	123	DH36:	.ASCIZ	/TEST #	ERR PC	PORT #/			
20	077756	124	105	123	DH42:	.ASCIZ	/TEST #	ERR PC/				
21	077775	040	040	040	DH44:	.ASCII	/		RELSNG	ERROR/<CR><LF>		
22	100034	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #	PORT #/		
23	100073	040	040	040	DH46:	.ASCII	/		PORT A	PORT B/<CR><LF>		
24	100133	124	105	123		.ASCIZ	/TEST #	ERR PC	RMDS	RMDS/		
25	100170	124	105	123	DH55:	.ASCIZ	/TEST #	ERR PC	PORT #	TIMEOUT VALUE (IN MS)/		
26	100246	044	122	115	DH56:	.ASCIZ	/\$RMADR/					
27	100255	124	105	123	DH57:	.ASCII	/TEST #	ERR PC	PORT A	PORT B/		
28	100327	040	040	040		.ASCIZ	/		EXPCTD	RECEVD	EXPECTD	RECEVD/
29												
30												
31						.EVEN						
32	100410	001246	001116	001240	DT1:	.WORD	TSTNUM,\$ERRPC,PTNBR,\$BDADR,\$BDDAT,0					
33	100424	001246	001116	001122	DT3:	.WORD	TSTNUM,\$ERRPC,\$BDADR,\$GDDAT,\$BDDAT,0					
34	100440	001246	001116	001240	DT5:	.WORD	TSTNUM,\$ERRPC,PTNBR,\$BDADR,\$GDDAT,\$BDDAT,0					
35	100456	001246	001116	001244	DT6:	.WORD	TSTNUM,\$ERRPC,OPPRT,\$BDADR,\$BDDAT,0					
36	100472	001246	001116	001242	DT7:	.WORD	TSTNUM,\$ERRPC,SEIZPT,PTNBR,\$BDADR,\$GDDAT,\$BDDAT,0					
37	100512	001246	001116	001242	DT13:	.WORD	TSTNUM,\$ERRPC,SEIZPT,PTNBR,\$BDADR,\$BDDAT,0					
38	100530	001246	001116	001242	DT22:	.WORD	TSTNUM,\$ERRPC,SEIZPT,PTNBR,0					
39	100542	001246	001116	001242	DT23:	.WORD	TSTNUM,\$ERRPC,SEIZPT,\$BDADR,\$BDDAT,0					
40	100556	001246	001116	001242	DT31:	.WORD	TSTNUM,\$ERRPC,SEIZPT,OPPRT,0					
41	100570	001246	001116	001242	DT36:	.WORD	TSTNUM,\$ERRPC,SEIZPT,0					
42	100600	001246	001116	001240	DT37:	.WORD	TSTNUM,\$ERRPC,PTNBR,0					
43	100610	001246	001116	000000	DT42:	.WORD	TSTNUM,\$ERRPC,0					
44	100616	001246	001116	001170	DT46:	.WORD	TSTNUM,\$ERRPC,\$TMP2,\$TMP3,0					
45	100630	001246	001116	001244	DT54:	.WORD	TSTNUM,\$ERRPC,OPPRT,SEIZPT,0					
46	100642	001246	001116	001242	DT55:	.WORD	TSTNUM,\$ERRPC,SEIZPT,TIME,0					
47	100654	001304	000000		DT56:	.WORD	\$RMADR,0					
48	100660	001246	001116	001164	DT57:	.WORD	TSTNUM,\$ERRPC,\$TMP0,\$TMP1,\$TMP2,\$TMP3,0					
49												
50	100676	000	000	000	DF1:	.BYTE	0,0,0,0,0					
51	100703	000	000	000	DF5:	.BYTE	0,0,0,0,0,0					
52	100711	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0					
53	100720	000	000	000	DF31:	.BYTE	0,0,0,0					
54	100724	000	000	000	DF36:	.BYTE	0,0,0					
55	100727	000	000	000	DF42:	.BYTE	0,0					
56	100731	000	000	000	DF55:	.BYTE	0,0,0,1					
57	100735	000			DF56:	.BYTE	0					

58  
59  
60

.EVEN



```

1      .SBTTL  CONSTANTS, TABLES, ETC
2
3      ;TABLE OF TEST STARTING ADDRESSES
4
5      TSTADR: .WORD  TST1      ;STARTING ADDRESS OF TEST 1
6      100736 003376      ;STARTING ADDRESS OF TEST 2
7      100740 004750      ;STARTING ADDRESS OF TEST 3
8      100742 006316      ;STARTING ADDRESS OF TEST 4
9      100744 007664      ;STARTING ADDRESS OF TEST 5
10     100746 011014      ;STARTING ADDRESS OF TEST 6
11     100750 012144      ;STARTING ADDRESS OF TEST 7
12     100752 012616      ;STARTING ADDRESS OF TEST 10
13     100754 013270      ;STARTING ADDRESS OF TEST 11
14     100756 014534      ;STARTING ADDRESS OF TEST 12
15     100760 016000      ;STARTING ADDRESS OF TEST 13
16     100762 017120      ;STARTING ADDRESS OF TEST 14
17     100764 020240      ;STARTING ADDRESS OF TEST 15
18     100766 021640      ;STARTING ADDRESS OF TEST 16
19     100770 023240      ;STARTING ADDRESS OF TEST 17
20     100772 024164      ;STARTING ADDRESS OF TEST 20
21     100774 025110      ;STARTING ADDRESS OF TEST 21
22     100776 026156      ;STARTING ADDRESS OF TEST 22
23     101000 027224      ;STARTING ADDRESS OF TEST 23
24     101002 031300      ;STARTING ADDRESS OF TEST 24
25     101004 032024      ;STARTING ADDRESS OF TEST 25
26     101006 033220      ;STARTING ADDRESS OF TEST 26
27     101010 034414      ;STARTING ADDRESS OF TEST 27
28     101012 036110      ;STARTING ADDRESS OF TEST 30
29     101014 037604      ;STARTING ADDRESS OF TEST 31
30     101016 041300      ;STARTING ADDRESS OF TEST 32
31     101020 042774      ;STARTING ADDRESS OF TEST 33
32     101022 044252      ;STARTING ADDRESS OF TEST 34
33     101024 045376      ;STARTING ADDRESS OF TEST 35
34     101026 046274      ;STARTING ADDRESS OF TEST 36
35     101030 047172      ;STARTING ADDRESS OF TEST 37
36     101032 050170      ;STARTING ADDRESS OF TEST 40
37     101034 051166      ;STARTING ADDRESS OF TEST 41
38     101036 053152      ;STARTING ADDRESS OF TEST 42
39     101040 055136      ;STARTING ADDRESS OF TEST 43
40     101042 056412      ;STARTING ADDRESS OF TEST 44
41     101044 057666      ;STARTING ADDRESS OF TEST 45
42     101046 060516      ;STARTING ADDRESS OF TEST 46
43     101050 061346
44
45     ;ATTENTION BIT TABLE
46
47     ATABIT: .BYTE  1      ;ATTENTION BIT FOR DRIVE 0
48     .BYTE  2      ;ATTENTION BIT FOR DRIVE 1
49     .BYTE  4      ;ATTENTION BIT FOR DRIVE 2
50     .BYTE 10      ;ATTENTION BIT FOR DRIVE 3
51     .BYTE 20      ;ATTENTION BIT FOR DRIVE 4
52     .BYTE 40      ;ATTENTION BIT FOR DRIVE 5
53     .BYTE 100     ;ATTENTION BIT FOR DRIVE 6
54     .BYTE 200     ;ATTENTION BIT FOR DRIVE 7
55
56     MAXTN: .WORD  52      ;MAXIMUM TEST NUMBER
57
58     .END  200
59
60
61

```



ADDRIS	072012	CRLF	= 000200	DT46	100616	EM57	076135	OPPRT	001244
ADRERR	071601	DCK	= 100000	DT5	100440	EM6	072322	OR	= 000200
AOE	= 001000	DDISP	= 177570	DT54	100630	EM60	076176	PAR	= 000010
ASR1	001236	DF1	100676	DT55	100642	EM61	076242	PAT	= 000020
ATA	= 100000	DF31	100720	DT56	100654	EM62	076274	PGE	= 002000
ATABIT	101052	DF36	100724	DT57	100660	EM63	076342	PGM	= 001000
ATO	= 000001	DF42	100727	DT6	100456	EM64	076404	PIP	= 020000
AT1	= 000002	DF5	100703	DT7	100472	EM65	076453	PIRQ	= 177772
AT2	= 000004	DF55	100731	DVA	= 004000	EM7	072422	PIRQVE	= 000240
AT3	= 000010	DF56	100735	DVC	= 000200	ENTERA	071552	PORTA	001224
AT4	= 000020	DF7	100711	ECH	= 000100	ERR	= 040000	PORTAI	071623
AT5	= 000040	DH1	076520	ECI	= 004000	ERROR	= 104000	PORTB	001226
AT6	= 000100	DH11	077162	EMTVEC	= 000030	ERRVEC	= 000004	PORTBI	071650
AT7	= 000200	DH13	077233	EM1	072076	EXEC	003110	PORTC	001230
A16	= 000400	DH22	077353	EM10	072503	FER	= 000020	PRO	= 000000
A17	= 001000	DH23	077451	EM11	072533	FMT16	= 010000	PR1	= 000040
BADNO	071753	DH26	077551	EM12	072617	F0	= 000002	PR2	= 000100
BADTMO	002160	DH3	076571	EM13	072667	F1	= 000004	PR3	= 000140
BAI	= 000010	DH31	077630	EM14	072754	F2	= 000010	PR4	= 000200
BIT0	= 000001	DH36	077727	EM15	073017	F3	= 000020	PR5	= 000240
BIT00	= 000001	DH4	076640	EM16	073076	F4	= 000040	PR6	= 000300
BIT01	= 000002	DH42	077756	EM17	073151	GO	= 000001	PR7	= 000340
BIT02	= 000004	DH44	077775	EM2	072117	GTSWR	= 104406	PS	= 177776
BIT03	= 000010	DH46	100073	EM20	073230	HCE	= 000200	PSEL	= 002000
BIT04	= 000020	DH5	076763	EM21	073310	HCI	= 002000	PSW	= 177776
BIT05	= 000040	DH55	100170	EM22	073363	HCRC	= 000400	PTNBR	001240
BIT06	= 000100	DH56	100246	EM23	073450	HT	= 000011	PWRVEC	= 000024
BIT07	= 000200	DH57	100255	EM24	073516	IAE	= 002000	RDCHR	= 104410
BIT08	= 000400	DH7	077037	EM25	073575	IE	= 000100	RDLIN	= 104411
BIT09	= 001000	DISPLA	001142	EM26	073640	ILF	= 000001	RDOCT	= 104412
BIT1	= 000002	DISPRE	000174	EM27	073725	ILR	= 000002	RDY	= 000200
BIT10	= 002000	DLT	= 100000	EM3	072141	IOTVEC	= 000020	RELERR	001254
BIT11	= 004000	DMD	= 000001	EM30	074003	IR	= 000100	RELOK	= 000001
BIT12	= 010000	DPE	= 000010	EM31	074100	IVC	= 010000	RESREG	= 104414
BIT13	= 020000	DPR	= 000400	EM32	074155	KYBCTL	001300	RESVEC	= 000010
BIT14	= 040000	DRQ	= 004000	EM33	074226	LBC	= 002000	RMAS	= 000016
BIT15	= 100000	DRY	= 000200	EM34	074330	LBT	= 002000	RMBA	= 000004
BIT2	= 000004	DSWR	= 177570	EM35	074433	LF	= 000012	RMCS1	= 000000
BIT3	= 000010	DTE	= 010000	EM36	074512	LSC	= 004000	RMCS2	= 000010
BIT4	= 000020	DT00	= 000001	EM37	074564	MAXTN	101062	RMDA	= 000006
BIT5	= 000040	DT01	= 000002	EM4	072223	MCPE	= 020000	RMDB	= 000022
BIT6	= 000100	DT02	= 000004	EM40	074632	MDPE	= 000400	RMDC	= 000034
BIT7	= 000200	DT03	= 000010	EM41	074707	MOH	= 020000	RMDS	= 000012
BIT8	= 000400	DT04	= 000020	EM42	074751	MOL	= 010000	RMDT	= 000026
BIT9	= 001000	DT05	= 000040	EM43	075037	MUR	= 001000	RMEC1	= 000044
BPTVEC	= 000014	DT06	= 000100	EM44	075114	MXF	= 001000	RMEC2	= 000046
CHANGE	003262	DT07	= 000200	EM45	075171	NBA	= 100000	RMER1	= 000014
CHGADR	001302	DT08	= 000400	EM46	075250	NED	= 010000	RMER2	= 000042
CKCLK	066124	DT1	100410	EM47	075317	NEM	= 004000	RMLA	= 000020
CKCLK1	066174	DT13	100512	EM5	072254	NOATA	= 000001	RMMR1	= 000024
CKCLK2	066236	DT22	100530	EM50	075405	NOCLOC	071675	RMMR2	= 000040
CKCLK3	066246	DT23	100542	EM51	075467	NOSEIZ	001252	RMOF	= 000032
CKERR	001250	DT3	100424	EM52	075556	NTRH11	072047	RMR	= 000004
CKSWR	= 104407	DT31	100556	EM53	075651	OFD	= 000200	RMSN	= 000030
CLOCK	066256	DT36	100570	EM54	075732	OM	= 000001	RMWC	= 000002
CLR	= 000040	DT37	100600	EM55	076025	OPE	= 020000	RQA	= 100000
CR	= 000015	DT42	100610	EM56	076072	OPI	= 020000	RQB	= 040000



RQSTA	001232	TEST16	023276	TST14	020240	\$BDADR	001122	SQUES	001206
RQSTB	001234	TEST17	024222	TST15	021640	\$BDDAT	001126	\$RDCHR	070640
R6	=%000006	TEST2	005006	TST16	023240	\$BELL	001202	\$RDLIN	070730
R7	=%000007	TEST20	025146	TST17	024164	\$CHARC	067406	\$RDOCT	071230
SAVREG	= 104413	TEST21	026214	TST2	004750	\$CKSWR	070336	\$RDSZ	= 000007
SC	= 100000	TEST22	027262	TST20	025110	\$CMTAG	001100	\$REGAD	001160
SCOPE	= 000004	TEST23	031336	TST21	026156	\$CM1	= 000001	\$REGO	001162
SCO	= 000100	TEST24	032062	TST22	027224	\$CM2	= 000002	\$RESRE	071426
SC1	= 000200	TEST25	033256	TST23	031300	\$CM3	= 000001	\$RMADR	001304
SC2	= 000400	TEST26	034452	TST24	032024	\$CM4	= 000005	\$RMVEC	001306
SC3	= 001000	TEST27	036146	TST25	033220	\$CNTLG	071200	\$RTNAD	066116
SC4	= 002000	TEST3	006354	TST26	034414	\$CNTLU	071173	\$SAVRE	071370
SEIZPT	001242	TEST30	037642	TST27	036110	\$CRLF	001207	\$SCOPE	066360
SKI	= 100000	TEST31	041336	TST3	006316	\$DBLK	070054	\$SETUP	= 000127
STACK	= 001100	TEST32	043032	TST30	037604	\$DOAGN	066114	\$STUP	= 177777
START	002240	TEST33	044310	TST31	041300	\$DTBL	070044	\$SVLAD	066546
START1	002246	TEST34	045434	TST32	042774	\$ENDAD	066104	\$SVPC	= 000210
START2	002254	TEST35	046332	TST33	044252	\$ENDCT	065750	\$SWR	= 166000
STKLMT	= 177774	TEST36	047230	TST34	045376	\$ENULL	066120	\$SWRMK	= 000000
SWR	001140	TEST37	050226	TST35	046274	\$EOP	065704	\$TIMES	001176
SWREG	000176	TEST4	007722	TST36	047172	\$EOPCT	065742	\$TKB	001146
SW0	= 000001	TEST40	051224	TST37	050170	\$ERFLG	001103	\$TKCNT	070064
SW00	= 000001	TEST41	053210	TST4	007664	\$ERMAX	001115	\$TKINT	070074
SW01	= 000002	TEST42	055174	TST40	051166	\$ERROR	066612	\$TKQEN	= 070073
SW02	= 000004	TEST43	056450	TST41	053152	\$ERRPC	001116	\$TKQIN	070066
SW03	= 000010	TEST44	057724	TST42	055136	\$ERRTB	001310	\$TKQOU	070070
SW04	= 000020	TEST45	060554	TST43	056412	\$ERRTY	066744	\$TKQSR	070072
SW05	= 000040	TEST46	061404	TST44	057666	\$ERTTL	001112	\$TKS	001144
SW06	= 000100	TEST47	062270	TST45	060516	\$ESCAP	001200	\$TKSRV	070144
SW07	= 000200	TEST5	011052	TST46	061346	\$FILLC	001156	\$TMP0	001164
SW08	= 000400	TEST50	063154	TST47	062232	\$FILLS	001155	\$TMP1	001166
SW09	= 001000	TEST51	064446	TST5	011014	\$GDADR	001120	\$TMP2	001170
SW1	= 000002	TEST6	012202	TST50	063116	\$GDDAT	001124	\$TMP3	001172
SW10	= 002000	TEST7	012654	TST50B	064166	\$GET42	066074	\$TMP4	001174
SW11	= 004000	TIME	001256	TST51	064410	\$GTSWR	070426	\$TN	= 000053
SW12	= 010000	TIMEA	001262	TST51B	065460	\$HD	= 000000	\$TPB	001152
SW13	= 020000	TIMEAM	001266	TST52	065702	\$HIOCT	071366	\$TPFLG	001157
SW14	= 040000	TIMEAP	001264	TST6	012144	\$ICNT	001104	\$TPS	001150
SW15	= 100000	TIMEB	001270	TST7	012616	\$INTAG	001135	\$TRAP	071464
SW2	= 000004	TIMEBM	001274	TYPDS	= 104405	\$ITEMB	001114	\$TRAP2	071506
SW3	= 000010	TIMEBP	001272	TYPE	= 104401	\$LF	001210	\$TRP	= 000015
SW4	= 000020	TIMES	001276	TYPOC	= 104402	\$LKCSB	001214	\$TRPAD	071520
SW5	= 000040	TKVEC	= 000060	TYPON	= 104404	\$LKCSR	001212	\$TSTNM	001102
SW6	= 000100	TOLER	066320	TYPOS	= 104403	\$LKS	001220	\$TTYIN	071164
SW7	= 000200	TPVEC	= 000064	UNS	= 040000	\$LLVEC	001222	\$TYPDS	067640
SW8	= 000400	TRAPVE	= 000034	UPE	= 020000	\$LPADR	001106	\$TYPE	067120
SW9	= 001000	TRE	= 040000	U0	= 000001	\$LPERR	001110	\$TYPEC	067270
TAP	= 040000	TRTVEC	= 000014	U1	= 000002	\$LPVEC	001216	\$TYPEX	067410
TBITVE	= 000014	TSTADR	100736	U3	= 000004	\$MNEW	071216	\$TYPOC	067436
TESTNO	071733	TSTERR	072001	VV	= 000100	\$MSWR	071205	\$TYPON	067452
TEST1	003434	TSTNUM	001246	VVSET	= 000001	\$MXCNT	066610	\$TYPOS	067412
TEST10	013326	TST1	003376	WATCH	001260	\$NULL	001154	\$XOFF	= 000023
TEST11	014572	TST1AA	003372	WCE	= 040000	\$NWTST	= 000000	\$XON	= 000021
TEST12	016036	TST10	013270	WCF	= 000040	\$OCNT	067634	\$XTSTR	066372
TEST13	017156	TST11	014534	WLE	= 004000	\$OMODE	067636	\$GET4	= 000000
TEST14	020276	TST12	016000	WRL	= 004000	\$OVER	066574	\$OFILL	067635
TEST15	021676	TST13	017120	\$AUTOB	001134	\$PASS	001100		

CZRMRA0 RM05/3/2 DU POR TST 1 MACRO V03.01 11-APR-80 14:26:10 PAGE 14-3  
SYMBOL TABLE

SEQ 0190

. ABS. 101064 000  
000000 001  
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 52712 WORDS ( 206 PAGES)  
DYNAMIC MEMORY AVAILABLE FOR 69 PAGES  
CZRMRA.BIN,CZRMRA/C=CZRMRA.DOC,CZRMRA,SYSMAC/M











SERTTL 5-0# 8-A92 8-A92\* 10-5 10-5 10-5\* L 15

SEQ 0193





8-571\* 8-623\* 8-639\* 8-664\* 8-684\* 8-720\* 8-739\*<sup>N 15</sup> 8-808\* 8-845\* 8-896\* 8-922\* 8-976\* 8-999\* 8-:54\*  
SEQ 0195

	8-:72*	8-:27*	8-:87*	8-<22*	8-<37*	8-<39*	8-<58*	8-=06*	8-=60*	8->26*	8->48*	8-?16*	8-?36*	8-?90*
\$LPVEC	8-@15*	8-@80*	8-A09*	8-A77*	10-1	10-1	10-1	10-1*						
\$MAIL	6-0#	9-11												
\$MNEW	8-24	8-30	10-1	10-5	10-7									
\$MSWR	10-10	10-10#												
\$MXCNT	10-10	10-10#												
\$NULL	10-1	10-1	10-1	10-1#										
\$NWTST	5-0#	10-7	10-7	10-7										
	8-139	8-139	8-139#	8-139#	8-201	8-201	8-201#	8-201#	8-290	8-290	8-290#	8-290#	8-308	8-308
	8-308#	8-308#	8-358	8-358	8-358#	8-358#	8-375	8-375	8-375#	8-375#	8-398	8-398	8-398#	8-398#
	8-421	8-421	8-421#	8-421#	8-467	8-467	8-467#	8-467#	8-491	8-491	8-491#	8-491#	8-548	8-548
	8-548#	8-548#	8-571	8-571	8-571#	8-571#	8-623	8-623	8-623#	8-623#	8-639	8-639	8-639#	8-639#
	8-664	8-664	8-664#	8-664#	8-684	8-684	8-684#	8-684#	8-720	8-720	8-720#	8-720#	8-739	8-739
	8-739#	8-739#	8-808	8-808	8-808#	8-808#	8-845	8-845	8-845#	8-845#	8-896	8-896	8-896#	8-896#
	8-922	8-922	8-922#	8-922#	8-976	8-976	8-976#	8-976#	8-999	8-999	8-999#	8-999#	8-:54	8-:54
	8-:54#	8-:54#	8-:72	8-:72	8-:72#	8-:72#	8-:27	8-:27	8-:27#	8-:27#	8-:87	8-:87	8-:87#	8-:87#
	8-<37	8-<37	8-<37#	8-<37#	8-<58	8-<58	8-<58#	8-<58#	8-=06	8-=06	8-=06#	8-=06#	8-=60	8-=60
	8-=60#	8-=60#	8->26	8->26	8->26#	8->26#	8->48	8->48	8->48#	8->48#	8-?16	8-?16	8-?16#	8-?16#
	8-?36	8-?36	8-?36#	8-?36#	8-?90	8-?90	8-?90#	8-?90#	8-@15	8-@15	8-@15#	8-@15#	8-@80	8-@80
	8-@80#	8-@80#	8-A09	8-A09	8-A09#	8-A09#	8-A77	8-A77	8-A77#	8-A77#	8-A84#			
\$SOCNT	10-8#	10-8*	10-8*											
\$SOMODE	10-8	10-8#	10-8*	10-8*	10-8*	10-8*								
\$SOVER	10-1	10-1	10-1	10-1#										
\$PASS	5-0#	8-77*	8-A92	8-A92	8-A92	8-A92*	8-A92*	10-1	10-1	10-1				
\$QUES	5-0#	10-5	10-5	10-7	10-7	10-10	10-10	10-10	10-10	10-11	10-11	10-11		
\$R2A	10-13													
\$RDCHR	10-10#	10-13	10-13											
\$RDDEC	10-13													
\$RDLIN	10-10#	10-13	10-13											
\$RDOCT	10-11#	10-13	10-13											
\$RDSZ	10-10	10-10#												
\$REGO	5-0#													
\$REGAD	5-0#													
\$RESRE	10-12#	10-13												
\$RMADR	6-0#	8-71	8-104	8-111*	8-114	8-137	13-47							
\$RMVEC	6-0#													
\$RTNAD	8-A92#													
\$SAVRE	10-12#	10-13	10-13											
\$SCOPE	8-24	10-1#												
\$SETUP	4-792	4-792	4-792	4-792	4-792	4-792#	4-792#	4-792#	4-792#	4-792#	4-792#	8-24	8-24	8-24
	8-24	8-24	8-24	8-24	8-24	8-24	8-24	8-24	8-24	8-30	8-30	8-30	8-A92	8-A92
	10-1	10-5	10-5	10-5	10-5	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10
\$STUP	4-792	4-792	4-792	4-792	4-792	4-792#	4-792#	4-792#	4-792#	4-792#	4-792#	4-792#	4-792#	4-792#
	4-792#													
\$SVLAD	10-1	10-1#												
\$SVPC	4-798	4-798#												
\$SWR	4-598#	4-607	4-608	4-608	4-608	4-608	4-608	4-608	4-608	4-608	5-0	5-0	5-0	8-24
	8-24	8-24	8-24	8-24	8-139	8-201	8-290	8-308	8-358	8-375	8-398	8-421	8-467	8-491
	8-548	8-571	8-623	8-639	8-664	8-684	8-720	8-739	8-808	8-845	8-896	8-922	8-976	8-999
	8-:54	8-:72	8-:27	8-:87	8-<37	8-<58	8-=06	8-=60	8->26	8->48	8-?16	8-?36	8-?90	8-@15
	8-@80	8-A09	8-A77	8-A84	8-A92	8-A92	8-A92	8-A92	8-A92	10-1	10-1	10-1	10-1	10-1
	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1	10-1
	10-3#	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5	10-5
\$SWRMK	10-1													
\$TIMES	5-0#	8-24*	8-139*	8-201*	8-290*	8-308*	8-358*	8-375*	8-398*	8-421*	8-467*	8-491*	8-548*	8-571*
	8-623*	8-639*	8-664*	8-684*	8-720*	8-739*	8-808*	8-845*	8-896*	8-922*	8-976*	8-999*	8-:54*	8-:72*



8-;27\* 8-;87\* 8-<37\* 8-<58\* 8-=06\* 8-=60\* 8->26\*<sup>C 16</sup> 8->48\* 8-?16\* 8-?36\* 8-?90\* 8-a15\* 8-a80\* 8-A09\*  
SEQ 0197



	8-A77*	8-A92*	10-1	10-1	10-1	10-1*	10-1*	10-10	10-10	10-10	10-10	10-10	10-10	10-10
\$TKB	5-0#	10-7	10-7	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10
\$TKCNT	10-10	10-10	10-10#	10-10*	10-10*	10-10*	10-10*	10-10	10-10	10-10	10-10	10-10	10-10	10-10
\$TKINT	8-32	8-75	10-10	10-10#										
\$TKGEN	10-10	10-10	10-10#											
\$TKQIN	10-10	10-10	10-10#	10-10*	10-10*	10-10*	10-10*							
\$TKQOU	10-10	10-10	10-10#	10-10*	10-10*	10-10*	10-10*							
\$TKQSR	10-10	10-10	10-10	10-10#										
\$TKS	5-0#	10-7	10-7	10-10	10-10	10-10	10-10	10-10	10-10	10-10*	10-10*	10-10*	10-10*	10-10*
\$TKSRV	10-10	10-10#												
\$TMP0	5-0#	8-152	8-152	8-152*	8-152*	8-152*	8-152*	8-167	8-167	8-167	8-167	8-167*	8-167*	8-167*
	8-167*	8-167*	8-167*	8-167*	8-167*	8-270	8-270	8-270	8-270	8-270*	8-270*	8-270*	8-270*	8-292
	8-292	8-292	8-292	8-292*	8-292*	8-292*	8-292*	8-342	8-342	8-342	8-342	8-342	8-342	8-342*
	8-342*	8-342*	8-342*	8-342*	8-342*	8-342*	8-342*	8-360	8-360	8-360	8-360	8-360	8-360	8-360*
	8-360*	8-360*	8-360*	8-360*	8-360*	8-360*	8-360*	8-387	8-387	8-387	8-387*	8-387*	8-400	8-400
	8-400	8-400*	8-400*	8-446	8-446	8-446	8-446	8-446	8-446	8-446*	8-446*	8-446*	8-446*	8-446*
	8-469	8-469	8-469	8-469	8-469	8-469	8-469*	8-469*	8-469*	8-469*	8-469*	8-469*	8-526	8-526
	8-526	8-526*	8-526*	8-526*	8-526*	8-550	8-550	8-550	8-550	8-550*	8-550*	8-550*	8-550*	8-602
	8-602	8-602	8-602	8-602	8-602	8-602	8-602	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*
	8-602*	8-602*	8-602*	8-602*	8-602*	8-625	8-625	8-625	8-625	8-625	8-625	8-625	8-625	8-625*
	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*
	8-650	8-650*	8-650*	8-650*	8-650*	8-666	8-666	8-666	8-666	8-666*	8-666*	8-666*	8-666*	8-702
	8-702	8-702	8-702	8-702	8-702*	8-702*	8-702*	8-702*	8-702*	8-702*	8-702*	8-702*	8-722	8-722
	8-722	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722*	8-722	8-722
	8-794	8-794	8-794	8-794	8-794	8-794*	8-794*	8-794*	8-794*	8-794*	8-794*	8-794*	8-794	8-794
	8-794*	8-824	8-824	8-824	8-824	8-824*	8-824*	8-824*	8-824*	8-824*	8-824*	8-824*	8-875	8-875
	8-875	8-875	8-875*	8-875*	8-875*	8-875*	8-875*	8-875*	8-875*	8-875*	8-875*	8-875*	8-875	8-875
	8-898	8-898	8-898*	8-898*	8-898*	8-898*	8-898*	8-898*	8-898*	8-898*	8-898*	8-898*	8-898	8-898
	8-956	8-956	8-956	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956	8-956
	8-956*	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978*	8-978*	8-978*	8-978*	8-978*
	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*
	8-:33	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33
	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*
	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08*
	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*
	8-:72*	8-:72*	8-:72*	8-:72*	8-:72*	8-:72*	8-:72*	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72*
	8-<39	8-<39	8-<39	8-<39*	8-<39*	8-<22	8-<22	8-<22	8-<22	8-<22*	8-<22*	8-<22*	8-<22*	8-<39
	8-<88*	8-<88*	8-<88*	8-<88*	8-08	8-08	8-08	8-08	8-08	8-08	8-08	8-08	8-08	8-08*
	8-08*	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03*
	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*
	8->03*	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28*
	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*
	8->28*	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*
	8-?00*	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*
	8-?18*	8-?72	8-?72	8-?72	8-?72*	8-?72*	8-?72*	8-?92	8-?92	8-?92	8-?92*	8-?92*	8-?92*	8-?92*
	8-@60*	8-@60*	8-@82	8-@82	8-@82	8-@82*	8-@82*	8-A49	8-A49	8-A49	8-A49	8-A49	8-A49	8-A49*
	8-A49*	8-A79	8-A79	8-A79	8-A79	8-A79	8-A79*	8-A79*	8-A79*	13-48	8-A49	8-A49	8-A49*	8-A49*
\$TMP1	5-0#	8-270	8-270	8-270	8-270*	8-270*	8-270*	8-270*	8-270*	8-292	8-292	8-292*	8-292*	8-292*
	8-292*	8-342	8-342	8-342*	8-342*	8-360	8-360	8-360*	8-360*	8-387	8-387	8-387*	8-387*	8-400
	8-400	8-400*	8-400*	8-446	8-446	8-446	8-446*	8-446*	8-446*	8-446*	8-446*	8-446*	8-446*	8-446*
	8-469*	8-469*	8-469*	8-526	8-526	8-526*	8-526*	8-550	8-550	8-550*	8-550*	8-550*	8-550*	8-602
	8-602*	8-602*	8-602*	8-602*	8-602*	8-625	8-625	8-625	8-625*	8-625*	8-625*	8-625*	8-625*	8-650
	8-650*	8-650*	8-650*	8-650*	8-650*	8-666	8-666	8-666	8-666*	8-666*	8-666*	8-666*	8-666*	8-702
	8-702*	8-702*	8-702*	8-702*	8-702*	8-722	8-722	8-722	8-722*	8-722*	8-722*	8-722*	8-722*	8-794
	8-794	8-794*	8-794*	8-794*	8-794*	8-824	8-824	8-824*	8-824*	8-824*	8-824*	8-824*	8-875	8-875
	8-898	8-898*	8-898*	8-956	8-956	8-956	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-978	8-978*



8-978\* 8-978\* 8-978\* 8-:33 8-:33 8-:33 8-:33<sup>E 16</sup>\* 8-:33\* 8-:33\* 8-:33\* 8-:56 8-:56 8-:56 8-:56\*  
SEQ 0199



	8-:56*	8-:56*	8-:56*	8-:08	8-:08	8-:08*	8-:08*	8-:72	8-:72	8-:72*	8-:72*	8-<22	8-<22	8-<22
	8-<22*	8-<22*	8-<22*	8-<22*	8-<39	8-<39	8-<39*	8-<39*	8-<39*	8-<39*	8-<39*	8-<88	8-<88	8-<88*
	8-<88*	8-=08	8-=08	8-=08*	8-=08*	8->03	8->03	8->03	8->03	8->03*	8->03*	8->03*	8->03*	8->03*
	8->28	8->28	8->28	8->28*	8->28*	8->28*	8->28*	8-?00	8-?00	8-?00*	8-?00*	8-?18	8-?18	8-?18*
	8-?18*	8-?72	8-?72	8-?72*	8-?72*	8-?92	8-?92	8-?92*	8-?92*	8-?92*	8-?92*	8-?18	8-?18	8-?18*
	8-a82	8-a82*	8-a82*	8-A49	8-A49	8-A49*	8-A49*	8-A79	8-A79	8-A79*	8-A79*	8-a60*	8-a60*	8-a82
\$TMP2	5-0#	8-270	8-270	8-270	8-270*	8-270*	8-270*	8-292	8-292	8-292	8-292*	13-48	8-342	8-342
	8-342*	8-342*	8-360	8-360	8-360	8-360*	8-360*	8-387	8-387	8-387	8-387*	8-387*	8-400	8-400
	8-400	8-400*	8-400*	8-446	8-446	8-446	8-446*	8-446*	8-469	8-469	8-469	8-469*	8-469*	8-526
	8-526	8-526	8-526*	8-526*	8-550	8-550	8-550*	8-550*	8-602	8-602	8-602	8-602*	8-602*	8-602*
	8-625	8-625	8-625	8-625*	8-625*	8-650	8-650	8-650*	8-650*	8-650*	8-650*	8-666	8-666	8-666*
	8-666*	8-702	8-702	8-702	8-702*	8-702*	8-702*	8-722	8-722	8-722	8-722*	8-722*	8-794	8-794
	8-794	8-794	8-794	8-794*	8-794*	8-794*	8-794*	8-824	8-824	8-824	8-824*	8-824*	8-875	8-875
	8-875	8-875*	8-875*	8-898	8-898	8-898	8-898*	8-898*	8-956	8-956	8-956	8-956*	8-956*	8-978
	8-978	8-978	8-978*	8-978*	8-:33	8-:33	8-:33	8-:33*	8-:33*	8-:33*	8-:33*	8-:56	8-:56	8-:56*
	8-:08	8-:08	8-:08	8-:08*	8-:08*	8-:72	8-:72	8-:72*	8-:72*	8-:72*	8-:72*	8-<22	8-<22	8-<22*
	8-<22*	8-<39	8-<39	8-<39	8-<39*	8-<39*	8-<39*	8-<88	8-<88	8-<88	8-<88*	8-<88*	8-=08	8-=08
	8-=08*	8-=08*	8->03	8->03	8->03	8->03	8->03	8->03	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*
	8->28	8->28	8->28	8->28	8->28*	8->28*	8->28*	8->28*	8-?00	8-?00	8-?00	8-?00*	8-?00*	8-?00*
	8-?18	8-?18	8-?18*	8-?18*	8-?72	8-?72	8-?72	8-?72*	8-?72*	8-?72*	8-?72*	8-?92	8-?92	8-?92*
	8-a60	8-a60	8-a60	8-a60*	8-a60*	8-a82	8-a82	8-a82*	8-a82*	8-a82*	8-a82*	8-A49	8-A49	8-A49*
	8-A49*	8-A79	8-A79	8-A79	8-A79*	8-A79*	8-A79*	13-44	13-48					
\$TMP3	5-0#	8-270	8-270	8-270	8-270*	8-270*	8-270*	8-292	8-292	8-292	8-292*	8-292*	8-342	8-342
	8-342*	8-342*	8-360	8-360	8-360	8-360*	8-360*	8-387	8-387	8-387	8-387*	8-387*	8-400	8-400
	8-400	8-400*	8-400*	8-446	8-446	8-446	8-446*	8-446*	8-469	8-469	8-469	8-469*	8-469*	8-526
	8-526	8-526	8-526*	8-526*	8-550	8-550	8-550*	8-550*	8-602	8-602	8-602	8-602*	8-602*	8-602*
	8-625	8-625	8-625	8-625*	8-625*	8-650	8-650	8-650*	8-650*	8-650*	8-650*	8-666	8-666	8-666*
	8-666*	8-702	8-702	8-702	8-702*	8-702*	8-702*	8-722	8-722	8-722	8-722*	8-722*	8-794	8-794
	8-794	8-794	8-794	8-794*	8-794*	8-794*	8-794*	8-824	8-824	8-824	8-824*	8-824*	8-875	8-875
	8-875	8-875*	8-875*	8-898	8-898	8-898	8-898*	8-898*	8-956	8-956	8-956	8-956*	8-956*	8-978
	8-978	8-978	8-978*	8-978*	8-:33	8-:33	8-:33	8-:33*	8-:33*	8-:33*	8-:33*	8-:56	8-:56	8-:56*
	8-:08	8-:08	8-:08	8-:08*	8-:08*	8-:72	8-:72	8-:72*	8-:72*	8-:72*	8-:72*	8-<22	8-<22	8-<22*
	8-<22*	8-<39	8-<39	8-<39	8-<39*	8-<39*	8-<39*	8-<88	8-<88	8-<88	8-<88*	8-<88*	8-=08	8-=08
	8-=08*	8-=08*	8->03	8->03	8->03	8->03	8->03	8->03	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*
	8->28	8->28	8->28	8->28	8->28*	8->28*	8->28*	8->28*	8-?00	8-?00	8-?00	8-?00*	8-?00*	8-?00*
	8-?18	8-?18	8-?18*	8-?18*	8-?72	8-?72	8-?72	8-?72*	8-?72*	8-?72*	8-?72*	8-?92	8-?92	8-?92*
	8-a60	8-a60	8-a60	8-a60*	8-a60*	8-a82	8-a82	8-a82*	8-a82*	8-a82*	8-a82*	8-A49	8-A49	8-A49*
	8-A49*	8-A79	8-A79	8-A79	8-A79*	8-A79*	8-A79*	13-44	13-48					
\$TMP4	5-0#	8-152	8-152	8-152*	8-152*	8-152*	8-152*	8-167	8-167	8-167	8-167*	8-167*	8-167*	8-167*
	8-167*	8-167*	8-167*	8-167*	8-167*	8-342	8-342	8-342	8-342*	8-342*	8-342*	8-342*	8-342*	8-342*
	8-360	8-360	8-360	8-360*	8-360*	8-360*	8-360*	8-360*	8-526	8-526	8-526*	8-526*	8-526*	8-550
	8-550*	8-602	8-602	8-602	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*	8-602*	8-625
	8-625	8-625	8-625	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-702	8-702*
	8-722	8-722*	8-722*	8-794	8-794	8-794*	8-794*	8-794*	8-794*	8-794*	8-794*	8-875	8-875	8-875*
	8-875*	8-898	8-898	8-898*	8-898*	8-898*	8-898*	8-898*	8-956	8-956	8-956	8-956*	8-956*	8-956*
	8-956*	8-956*	8-956*	8-956*	8-956*	8-978	8-978	8-978	8-978	8-978	8-978*	8-978*	8-978*	8-978*
	8-978*	8-978*	8-978*	8-:33	8-:33	8-:33	8-:33	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*
	8-:33*	8-:56	8-:56	8-:56	8-:56	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*
	8-:08	8-:08	8-:08	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*
	8-:72*	8-:72*	8-:72*	8-<88	8-<88	8-<88*	8-<88*	8-<88*	8-<88*	8-<88*	8-<88*	8-=08	8-=08	8-=08*
	8-=08*	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*
	8->03*	8->03*	8->03*	8->03*	8->03*	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28*	8->28*
	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8->28*	8-?00	8-?00	8-?00*
	8-?00*	8-?00*	8-?00*	8-?00*	8-?18	8-?18	8-?18	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*
\$TN	4-599#	4-607	8-139	8-139	8-139	8-139#	8-201	8-201	8-201	8-201#	8-290	8-290	8-290	8-290#
	8-308	8-308	8-308	8-308#	8-358	8-358	8-358	8-358#	8-375	8-375	8-375	8-375#	8-398	8-398



8-398 8-398# 8-421 8-421 8-421 8-421# 8-467<sup>G 16</sup> 8-467 8-467 8-467# 8-491 8-491 8-491 8-491#  
SEQ 0201



	8-548	8-548	8-548	8-548#	8-571	8-571	8-571	8-571#	8-623	8-623	8-623	8-623#	8-639	8-639
	8-639	8-639#	8-664	8-654	8-664	8-664#	8-684	8-684	8-684	8-684#	8-720	8-720	8-720	8-720#
	8-739	8-739	8-739	8-739#	8-808	8-808	8-808	8-808#	8-845	8-845	8-845	8-845#	8-896	8-896
	8-896	8-896#	8-922	8-922	8-922	8-922#	8-976	8-976	8-976	8-976#	8-999	8-999	8-999	8-999#
	8-:54	8-:54	8-:54	8-:54#	8-:72	8-:72	8-:72	8-:72#	8-:27	8-:27	8-:27	8-:27#	8-:87	8-:87
	8-:87	8-:87#	8-<37	8-<37	8-<37	8-<37#	8-<58	8-<58	8-<58	8-<58#	8-=06	8-=06	8-=06	8-=06#
	8-=60	8-=60	8-=60	8-=60#	8->26	8->26	8->26	8->26#	8->48	8->48	8->48	8->48#	8-?16	8-?16
	8-?16	8-?16#	8-?36	8-?36	8-?36	8-?36#	8-?90	8-?90	8-?90	8-?90#	8-@15	8-@15	8-@15	8-@15#
	8-@80	8-@80	8-@80	8-@80#	8-A09	8-A09	8-A09	8-A09#	8-A49	8-A50	8-A50	8-A77	8-A77	8-A77
	8-A77#	8-A79	8-A80	8-A80	8-A84	8-A84	8-A84#	14-27						
\$TPB	5-0#	10-7	10-7	10-7*										
\$TPFLG	5-0#	10-7	10-7	10-7										
\$TPS	5-0#	10-7	10-7	10-7										
\$TRAP	8-24	10-13#												
\$TRAP2	10-13	10-13#												
\$TRP	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13
	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13
	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13
	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13	10-13
	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#	10-13#
\$TRPAD	10-13	10-13#												
\$TSTNM	5-0#	8-139*	8-201*	8-290*	8-308*	8-358*	8-375*	8-398*	8-421*	8-467*	8-491*	8-548*	8-571*	8-623*
	8-639*	8-664*	8-684*	8-720*	8-739*	8-808*	8-845*	8-896*	8-922*	8-976*	8-999*	8-:54*	8-:72*	8-:27*
	8-:87*	8-<37*	8-<58*	8-=06*	8-=60*	8->26*	8->48*	8-?16*	8-?36*	8-?90*	8-@15*	8-@80*	8-A09*	8-A77*
	8-A92*	10-1	10-1	10-1	10-1	10-1*	10-5	10-5	10-5	10-5				
\$TTYIN	10-10	10-10	10-10	10-10	10-10	10-10#								
\$TYPBN	10-13													
\$TYPDS	10-9#	10-13	10-13											
\$TYPE	10-7#	10-13	10-13											
\$TYPEC	10-7	10-7	10-7	10-7#	10-10									
\$TYPEX	10-7	10-7	10-7#											
\$TYPOC	10-8#	10-13	10-13											
\$TYPON	10-8	10-8#	10-13											
\$TYPOS	10-8#	10-13												
\$XOFF	10-7	10-7												
\$XON	10-7	10-7												
\$XTSTR	10-1#													
A16	4-618#													
A17	4-619#													
ADDRIS	8-103	11-11#												
ADRERR	8-39	11-4#												
AOE	4-688#													
ASR1	6-0#	8-56*	8-794	8-824	8-:72	8-:72	8-:72	8-<22	8-<22	8-<39	8-<39	8-A49	8-A79	
AT0	4-705#													
AT1	4-706#													
AT2	4-707#													
AT3	4-708#													
AT4	4-709#													
AT5	4-710#													
AT6	4-711#													
AT7	4-712#													
ATA	4-675#	8-270	8-270	8-270	8-270	8-292	8-292	8-292	8-292	8-342	8-342	8-342	8-342	8-342
	8-342	8-342	8-342	8-360	8-360	8-360	8-360	8-360	8-360	8-360	8-360	8-387	8-387	8-387
	8-387	8-400	8-400	8-400	8-400	8-446	8-446	8-446	8-446	8-446	8-469	8-469	8-469	8-469
	8-469	8-526	8-526	8-526	8-526	8-550	8-550	8-550	8-550	8-602	8-602	8-602	8-602	8-625
	8-625	8-625	8-625	8-650	8-650	8-650	8-650	8-650	8-666	8-666	8-666	8-666	8-702	8-702



8-702 8-722 8-722 8-722 8-722 8-794 8-794<sup>1</sup> 16 8-794 8-794 8-794 8-794 8-794 8-794 8-794 8-794  
SEQ 0203

	8-794	8-824	8-824	8-824	8-824	8-875	8-875	8-875	8-875	8-875	8-875	8-875	8-875	8-898
	8-898	8-898	8-898	8-898	8-898	8-898	8-898	8-956	8-956	8-956	8-956	8-956	8-956	8-956
	8-956	8-956	8-956	8-956	8-956	8-956	8-956	8-956	8-978	8-978	8-978	8-978	8-978	8-978
	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-:33	8-:33	8-:33	8-:33	8-:33
	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:56	8-:56	8-:56	8-:56
	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:08	8-:08	8-:08
	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:72	8-:72	8-:72	8-:72
	8-:72	8-:72	8-:72	8-<22	8-<22	8-<22	8-<22	8-<39	8-<39	8-<39	8-<39	8-<88	8-<88	8-<88
	8-<88	8-<88	8-<88	8-<88	8-=08	8-=08	8-=08	8-=08	8-=08	8-=08	8-=08	8->03	8->03	8->03
	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->28	8->28	8->28
	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8-?00	8-?00	8-?00
	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18
	8-?18	8-?72	8-?72	8-?92	8-?92	8-?92	8-@60	8-@60	8-@60	8-@60	8-@82	8-@82	8-@82	8-A49
	8-A49	8-A49	8-A79	8-A79	8-A79									
ATABIT	8-56	14-18#												
BADNO	8-88	11-9#												
BADTMO	8-3#	8-26												
BAI	4-630#													
BIT0	4-610#													
BIT00	4-610	4-610#	4-665											
BIT01	4-610	4-610#												
BIT02	4-610	4-610#												
BIT03	4-610	4-610#												
BIT04	4-610	4-610#												
BIT05	4-610	4-610#												
BIT06	4-610	4-610#												
BIT07	4-610	4-610#												
BIT08	4-610	4-610#												
BIT09	4-610	4-610#	4-699	10-1										
BIT1	4-610#													
BIT10	4-610#	10-5												
BIT11	4-610#	10-1												
BIT12	4-610#													
BIT13	4-610#	10-5												
BIT14	4-610#	4-700	10-1											
BIT15	4-610#	4-701												
BIT2	4-610#													
BIT3	4-610#													
BIT4	4-610#													
BIT5	4-610#	4-661												
BIT6	4-610#													
BIT7	4-610#													
BIT8	4-610#													
BIT9	4-610#													
BPTVEC	4-610#													
CHANGE	8-33	8-100#												
CHGADR	6-0#	8-14*	8-17*	8-100	8-102*									
CKCLK	8-60	8-73	9-8#											
CKCLK1	9-8	9-17#												
CKCLK2	9-18	9-25#												
CKCLK3	9-16	9-24	9-27#											
CKERR	6-0#	8-152	8-152	8-152*	8-152*	8-152*	8-152*	8-159*	8-159*	8-159*	8-159*	8-167*	8-167*	8-167*
	8-167*	8-167*	8-167*	8-167*	8-167*	8-342*	8-342*	8-342*	8-342*	8-342*	8-342*	8-360*	8-360*	8-360*
	8-360*	8-360*	8-360*	8-446	8-446*	8-446*	8-446*	8-469	8-469*	8-469*	8-469*	8-526*	8-526*	8-526*
	8-526*	8-526*	8-550	8-550*	8-550*	8-550*	8-550*	8-550*	8-550*	8-550*	8-550*	8-602*	8-602*	8-602*
	8-602*	8-602*	8-602*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-625*	8-702*	8-702*	8-722*



8-722\* 8-794\* 8-794\* 8-794\* 8-794\* 8-875\* 8-875\*<sup>K 16</sup> 8-875\* 8-875\* 8-898\* 8-898\* 8-898\* 8-898\* 8-956  
SEQ 0205

	8-956	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-956*	8-978	8-978	8-978*	8-978*	8-978*
	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978*	8-978	8-978	8-978*	8-978*	8-978*
	8-:33*	8-:56	8-:56	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*	8-:33*	8-:33*	8-:33*	8-:33*	8-:33*
	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:08*	8-:72	8-:72	8-:72*	8-:72*	8-:72*
	8-<88*	8-<88*	8-<88*	8-08*	8-08*	8-08*	8-08*	8-08*	8-08*	8-08*	8->03*	8->03*	8->03*	8->03*	8->03*
	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->03*	8->28*	8->28*	8->28*	8->28*	8->28*
	8->28*	8->28*	8->28*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*	8-?00*
	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*	8-?18*
	8-A50*	8-A80*	8-A80*	8-A80*	8-A80*	8-A80*	8-A80*	8-A80*	8-A80*	8-A80*	8-a82	8-a82*	8-a82*	8-A50*	8-A50*
CKSWR	10-1	10-5	10-5	10-13#											
CLOCK	9-12	9-21	9-32#												
CLR	4-632#	4-661#	8-141	8-152	8-152	8-342	8-360	8-526	8-550	8-602	8-625	8-:08			
CR	4-610#	10-7	10-7	13-3	13-6	13-9	13-11	13-13	13-15	13-17	13-21	13-23			
CRLF	4-610#	8-6	8-30	8-30	10-7	10-7	11-3	11-4	11-5	11-6	11-7	11-7	11-9	11-10	
DCK	11-11														
DDISP	4-694#														
DF1	4-610#	5-0	8-24												
DF31	7-7	7-14	7-21	7-63	7-133	7-140	7-226	7-233	7-261	7-275	7-303	7-346	13-50#		
DF36	7-126	7-156	7-177	7-254	7-268	7-310	13-53#								
DF36	7-70	7-112	7-119	7-148	7-170	7-191	7-198	7-205	7-212	7-219	7-360	13-54#			
DF42	7-240	7-247	13-55#												
DF5	7-35	7-42	7-56	7-77	7-84	7-184	7-282	7-289	7-296	7-332	7-339	7-353	7-374	13-51#	
DF55	7-317	13-56#													
DF56	7-324	13-57#													
DF7	7-28	7-49	7-91	7-98	7-105	7-163	7-367	13-52#							
DH1	7-5	7-12	7-224	7-259	13-1#										
DH11	7-61	13-8#													
DH13	7-40	7-75	7-82	7-280	7-294	7-351	13-9#								
DH22	7-124	7-154	7-308	13-11#											
DH23	7-131	7-138	7-231	7-273	7-301	7-344	13-13#								
DH26	13-15#														
DH3	7-19	13-2#													
DH31	7-175	13-17#													
DH36	7-68	7-110	7-117	7-146	7-168	7-189	7-196	7-203	7-210	7-217	7-358	13-19#			
DH4	7-26	13-3#													
DH42	7-238	7-245	13-20#												
DH44	7-252	13-21#													
DH46	7-266	13-23#													
DH5	7-33	7-54	7-182	7-287	7-337	7-372	13-5#								
DH55	7-315	13-25#													
DH56	7-322	13-26#													
DH57	7-330	13-27#													
DH7	7-47	7-89	7-96	7-103	7-161	7-365	13-6#								
DISPLA	5-0#	8-24*	8-24*	10-1*	10-5*										
DISPRE	4-794#	8-24													
DLT	4-642#														
DMD	4-698#	8->03	8->03	8->28	8->28										
DPE	4-743#														
DPR	4-668#	8-167	8-167	8-270	8-270	8-292	8-292	8-342	8-360	8-387	8-400	8-446	8-446	8-446	
	8-469	8-469	8-469	8-526	8-550	8-602	8-602	8-602	8-602	8-625	8-625	8-625	8-625	8-650	
	8-650	8-666	8-666	8-702	8-702	8-722	8-722	8-794	8-794	8-794	8-794	8-794	8-794	8-824	
	8-875	8-875	8-898	8-898	8-956	8-956	8-978	8-978	8-:33	8-:33	8-:56	8-:56	8-<22	8-<39	
	8-<88	8-08	8->03	8->03	8->03	8->28	8->28	8->28	8-?72	8-?92	8-a60	8-a82	8-A49	8-A49	
	8-A79	8-A79													
DRQ	4-728#														
DRY	4-667#	8-167	8-167	8-270	8-270	8-292	8-292	8-342	8-360	8-387	8-400	8-446	8-446	8-446	



8-469 8-469 8-469 8-526 8-550 8-602 8-602<sup>B</sup> 1 8-602 8-602 8-625 8-625 8-625 8-625 8-625 8-650  
SEQ 0207





EM31 7-174 12-27#

D 1

SEQ 0209





HT

4-610# 10-7 10-7

F 1

SEQ 0211

IAE	4-689#													
IE	4-616#													
ILF	4-679#													
ILR	4-680#													
IOTVEC	4-610#	8-24*	8-24*											
IR	4-633#													
IVC	4-747#													
KYBCTL	6-0#	8-76*	8-93*	8-139	8-139*	8-201	8-201*	8-290	8-290*	8-308	8-308*	8-358	8-358*	8-375
	8-375*	8-398	8-398*	8-421	8-421*	8-467	8-467*	8-491	8-491*	8-548	8-548*	8-571	8-571*	8-623
	8-623*	8-639	8-639*	8-664	8-664*	8-684	8-684*	8-720	8-720*	8-739	8-739*	8-808	8-808*	8-845
	8-845*	8-896	8-896*	8-922	8-922*	8-976	8-976*	8-999	8-999*	8-:54	8-:54*	8-:72	8-:72*	8-:27
	8-:27*	8-:87	8-:87*	8-<37	8-<37*	8-<58	8-<58*	8-=06	8-=06*	8-=60	8-=60*	8->26	8->26*	8->48
	8->48*	8-?16	8-?16*	8-?36	8-?36*	8-?90	8-?90*	8-@15	8-@15*	8-@80	8-@80*	8-A09	8-A09*	8-A77
	8-A77*	8-A92												
LBC	4-745#													
LBT	4-670#													
LF	4-610#	10-7	10-7	11-7	11-8	11-11	11-12	13-3	13-6	13-9	13-11	13-13	13-15	13-17
	13-21	13-23												
LSC	4-746#													
MAXTN	8-86	14-29#												
MCPE	4-621#	8-446	8-469	8-A50	8-A50	8-A80	8-A80							
MDPE	4-635#													
MOH	4-729#													
MOL	4-672#	8-167	8-167	8-270	8-270	8-292	8-292	8-342	8-360	8-387	8-400	8-446	8-446	8-446
	8-469	8-469	8-469	8-526	8-550	8-602	8-602	8-602	8-602	8-625	8-625	8-625	8-625	8-650
	8-650	8-666	8-666	8-702	8-702	8-722	8-722	8-794	8-794	8-794	8-794	8-794	8-794	8-824
	8-875	8-875	8-898	8-898	8-956	8-956	8-978	8-978	8-:33	8-:33	8-:56	8-:56	8-:56	8-:39
	8-<88	8-=08	8->03	8->03	8->03	8->28	8->28	8->28	8-?7?	8-?92	8-@60	8-@82	8-A49	8-A49
	8-A79	8-A79												
MUR	4-699#	8->03	8->28											
MXF	4-636#													
NBA	4-731#													
NED	4-639#	8-152	8-152	8-152	8-152									
NEM	4-638#													
NOATA	7-379#	8-270	8-270	8-292	8-292	8-342	8-342	8-360	8-360	8-387	8-387	8-387	8-387	8-400
	8-400	8-400	8-400	8-446	8-446	8-446	8-446	8-469	8-469	8-469	8-469	8-526	8-526	8-526
	8-526	8-550	8-550	8-550	8-550	8-602	8-602	8-602	8-602	8-625	8-625	8-625	8-625	8-650
	8-650	8-650	8-650	8-666	8-666	8-666	8-666	8-702	8-702	8-702	8-702	8-722	8-722	8-722
	8-722	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-824	8-824	8-824	8-824	8-875
	8-875	8-875	8-875	8-875#	8-875#	8-898	8-898	8-898	8-898	8-898#	8-898#	8-900#	8-956	8-956
	8-956	8-956	8-978	8-978	8-978	8-978	8-:33	8-:33	8-:33	8-:33	8-:56	8-:56	8-:56	8-:56
	8-:08	8-:08	8-:08	8-:08	8-:08#	8-:10#	8-:72	8-:72	8-:72	8-:72	8-<22	8-<22	8-<22	8-<22
	8-<22#	8-<22#	8-<39	8-<39	8-<39	8-<39	8-<39#	8-<39#	8-<39#	8-<88	8-<88	8-<88	8-<88	8-=08
	8-=08	8-=08	8->03	8->03	8->03	8->03	8->03#	8->03#	8->28	8->28	8->28	8->28#	8->28#	8->30#
	8-?00	8-?00	8-?00	8-?00#	8-?00#	8-?18	8-?18	8-?18	8-?18	8-?18#	8-?18#	8-?72	8-?72	8-?72
	8-?72	8-?92	8-?92	8-?92	8-?92	8-@60	8-@60	8-@60	8-@60	8-@60#	8-@60#	8-@82	8-@82	8-@82
	8-@82	8-@82#	8-@82#	8-A49	8-A49	8-A49	8-A49	8-A49	8-A79	8-A79	8-A79	8-A79	8-A79	8-A79
NOCLOC	8-62	11-7#												
NOSEIZ	6-0#													
NTRH11	8-107	11-12#												
OFD	4-753#													
OM	4-665#	8-270	8-270	8-270	8-292	8-292	8-292	8-342	8-342	8-360	8-360	8-387	8-387	8-400
	8-400	8-446	8-446	8-446	8-469	8-469	8-469	8-526	8-526	8-550	8-550	8-602	8-602	8-602
	8-625	8-625	8-625	8-650	8-650	8-650	8-666	8-666	8-666	8-702	8-702	8-702	8-722	8-722
	8-722	8-794	8-794	8-794	8-794	8-824	8-824	8-824	8-824	8-875	8-875	8-898	8-898	8-956
	8-978	8-978	8-978	8-:33	8-:33	8-:33	8-:56	8-:56	8-:56	8-:72	8-:72	8-:72	8-:72	8-<22



8-<22 8-<39 8-<39 8-<88 8-<88 8-=08 8-=08<sup>H</sup> 1 8->03 8->03 8->03 8->03 8->28 8->28 8->28  
SEQ 0213





8-a60

8-a60

8-a60

8-a60

8-a60

8-a82

8-a82

<sup>J</sup> 1

8-a82

8-a82

8-a82

8-a82

8-a82

8-a82

8-a82  
SEQ 0215





8-602\* 8-602\* 8-625\* 8-625\* 8-625\* 8-625\* 8-625\*<sup>L</sup> 1 8-625\* 8-625\* 8-625\* 8-625\* 8-625\* 8-625\* 8-625\* 8-625\* 8-650\*  
SEQ 0217





8-?18\* 8-?18\* 8-?18\* 8-?18\* 8-?18\* 8-?72\* 8-?72\*<sup>N 1</sup> 8-?72\* 8-?72\* 8-?72\* 8-?72\* 8-?92\* 8-?92\* 8-?92\*

SEQ 0219





8-898\* 8-898\* 8-898\* 8-956 8-956 8-956 8-956<sup>c</sup> 2 8-956 8-956 8-956 8-956 8-956 8-956 8-956 8-956  
SEQ 0221

	8-956	8-956	8-956	8-956	8-956	8-956	8-956*	8-978	8-978	8-978	8-978	8-978	8-978	8-978
	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978	8-978*	8-:33	8-:33	8-:33
	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33	8-:33
	8-:33*	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56	8-:56
	8-:56	8-:56	8-:56	8-:56	8-:56*	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08	8-:08
	8-:08	8-:08	8-:08	8-:08	8-:08	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72	8-:72
	8-:72	8-:72	8-:72	8-<22	8-<22	8-<22	8-<22	8-<22*	8-<22*	8-<22*	8-<39	8-<39	8-<39	8-<39
	8-<39*	8-<39*	8-<39*	8-<88	8-<88	8-<88	8-<88	8-<88	8-<88	8-<88	8-<88*	8-<88*	8-<88*	8=<88*
	8-=08	8-=08	8-=08	8-=08	8-=08	8-=08	8-=08	8-=08*	8-=08*	8->03	8->03	8->03	8->03	8->03
	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03
	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28
	8->28	8->28	8->28	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00	8-?00*
	8-?00*	8-?00*	8-?00*	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18	8-?18*
	8-?18*	8-?18*	8-?18*	8-?72	8-?72	8-?72	8-?72	8-?72*	8-?72*	8-?72*	8-?72*	8-?72*	8-?72*	8-?72*
	8-?92	8-?92	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*	8-?92*
	8-@82	8-@82	8-@82	8-@82	8-@82	8-@82*	8-@82*	8-@82*	8-@82*	8-@82*	8-@82*	8-@82*	8-@82*	8-@82*
	8-A49	8-A49*	8-A49*	8-A49*	8-A49*	8-A79	8-A79	8-A79	8-A79	8-A79	8-A79*	8-A79*	8-A79*	8-A79*
RMDT	4-783#	8-159	8-159	8-159	8-159	8-270	8-270	8-292	8-292	8-292	8-292	8-292	8-292	8-292
RMEC1	4-789#	8-270	8-270	8-292	8-292									
RMEC2	4-790#	8-270	8-270	8-292	8-292									
RMER1	4-778#	8-270	8-270	8-292	8-292	8-526	8-526	8-526*	8-550	8-550	8-550*	8-956*	8-956*	8-956*
	8-956*	8-978*	8-978*	8-978*	8-978*	8-:33*	8-:33*	8-:33*	8-:33*	8-:56*	8-:56*	8-:56*	8-:56*	8-:56*
	8-:08*	8-:08*	8-:08*	8-:72*	8-:72*	8-:72*	8-:72*	8-<22*	8-<22*	8-<39*	8-<39*	8-?00	8-?00	8-?00*
	8-?18	8-?18	8-?18*	8-A49*	8-A49*	8-A49*	8-A50	8-A50	8-A79*	8-A79*	8-A79*	8-A80	8-A80	8-?00*
RMER2	4-788#	8-270	8-270	8-292	8-292	8-A49*	8-A49*	8-A49*	8-A50	8-A50	8-A79*	8-A79*	8-A79*	8-A80
	8-A80													
RPLA	4-780#	8-270	8-270	8-292	8-292									
RMRR1	4-782#	8-270	8-270	8-292	8-292	8->03*	8->03*	8->03*	8->28*	8->28*	8->28*			
RMRR2	4-787#													
RMOF	4-785#	8-270	8-270	8-292	8-292									
RMR	4-681#													
RMSN	4-784#	8-172	8-174	8-270	8-270	8-292	8-292							
RMWC	4-773#	8-115												
RQA	4-701#													
RQB	4-700#													
RQSTA	6-0#													
RQSTB	6-0#													
SAVREG	10-13#													
SC	4-623#													
SCO	4-735#													
SC1	4-736#													
SC2	4-737#													
SC3	4-738#													
SC4	4-739#													
SCOPE	4-610#	8-181	8-270	8-292	8-342	8-360	8-387	8-400	8-446	8-469	8-526	8-550	8-602	8-625
	8-650	8-666	8-702	8-722	8-794	8-824	8-875	8-898	8-956	8-978	8-:33	8-:56	8-:08	8-:72
	8-<22	8-<39	8-<88	8-=08	8->03	8->28	8-?00	8-?18	8-?72	8-?92	8-@60	8-@82	8-A49	8-A79
	8-A84													
SEIZPT	6-0#	8-270*	8-292*	8-342*	8-360*	8-387*	8-400*	8-446*	8-469*	8-526*	8-550*	8-602*	8-625*	8-650*
	8-666*	8-702*	8-722*	8-794*	8-794*	8-794*	8-875*	8-898*	8-956*	8-978*	8-:33*	8-:56*	8-<22*	8-<39*
	8-<88*	8-=08*	8->03*	8->28*	8-?00*	8-?18*	8-?72*	8-?92*	8-@60*	8-@82*	8-A49*	8-A79*	13-36	13-37
	13-38	13-39	13-40	13-41	13-45	13-46								
SKI	4-749#													
STACK	4-610#	8-24	8-72	8-139	8-201	8-290	8-308	8-358	8-375	8-398	8-421	8-467	8-491	8-548
	8-571	8-623	8-639	8-664	8-684	8-720	8-739	8-808	8-845	8-896	8-922	8-976	8-999	8-:54
	8-:72	8-:27	8-:87	8-<37	8-<58	8-=06	8-=60	8->26	8->48	8-?16	8-?36	8-?90	8-@15	8-@80



8-A09 8-A77

E 2

SEQ 0223





TEST32 8-:72 8-:72 8-:72#

6 2

SEQ 0225





TST31 8-:54# 14-11

1 2

SEQ 0227

TST32	8-:72#	14-11												
TST33	8-:27#	14-11												
TST34	8-:87#	14-11												
TST35	8-<37#	14-11												
TST36	8-<58#	14-11												
TST37	8-=06#	14-14												
TST4	8-308#	14-8												
TST40	8-=60#	14-14												
TST41	8->26#	14-14												
TST42	8->48#	14-14												
TST43	8-?16#	14-14												
TST44	8-?36#	14-14												
TST45	8-?90#	14-14												
TST46	8-@15#	14-14												
TST47	8-@80#													
TST5	8-358#	14-8												
TST50	8-A09#													
TST50B	8-A49	8-A49	8-A50#											
TST51	8-A50	8-A77#												
TST51B	8-A79	8-A79	8-A80#											
TST52	8-A80	8-A84#												
TST6	8-375#	14-8												
TST7	8-398#	14-8												
TSTADR	8-92	14-5#												
TSTERR	11-10#													
TSTNUM	6-0#	10-5*	13-32	13-33	13-34	13-35	13-36	13-37	13-38	13-39	13-40	13-41	13-42	13-43
	13-44	13-45	13-46	13-48										
TYPDS	8-A92	8-A92	10-6	10-13#										
TYPE	8-6	8-30	8-34	8-39	8-47	8-49	8-51	8-62	8-70	8-81	8-88	8-103	8-106	8-107
	8-A92	8-A92	8-A92	10-5	10-5	10-6	10-6	10-6	10-6	10-6	10-6	10-6	10-7	10-8
	10-9	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10	10-10
	10-10	10-10	10-10	10-11	10-11	10-13#								
TYPOC	8-8	8-105	10-6	10-6	10-10	10-13#								
TYPON	10-13#													
TYPOS	8-48	8-50	10-13#											
U0	4-627#													
U1	4-628#													
U3	4-629#													
UNS	4-693#													
UPE	4-640#													
VV	4-666#	8-270	8-270	8-270	8-270	8-292	8-292	8-292	8-292	8-342	8-342	8-342	8-342	8-342
	8-342	8-342	8-360	8-360	8-360	8-360	8-360	8-360	8-360	8-387	8-387	8-387	8-400	8-400
	8-400	8-446	8-446	8-446	8-446	8-446	8-469	8-469	8-469	8-469	8-469	8-526	8-526	8-526
	8-550	8-550	8-550	8-602	8-602	8-602	8-602	8-602	8-602	8-625	8-625	8-625	8-625	8-625
	8-625	8-650	8-650	8-650	8-650	8-666	8-666	8-666	8-666	8-702	8-702	8-702	8-702	8-722
	8-722	8-722	8-722	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-794	8-824
	8-824	8-824	8-875	8-875	8-875	8-875	8-898	8-898	8-898	8-898	8-956	8-956	8-956	8-956
	8-978	8-978	8-978	8-978	8-:33	8-:33	8-:33	8-:33	8-:33	8-:56	8-:56	8-:56	8-:56	8-:08
	8-:72	8-:72	8-<22	8-<22	8-<22	8-<39	8-<39	8-<39	8-<39	8-<88	8-<88	8-<88	8-<88	8-<88
	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03	8->03
	8->03	8->03	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28	8->28
	8->28	8->28	8->28	8->28	8-?00	8-?00	8-?18	8-?18	8-?18	8-?72	8-?72	8-?72	8-?92	8-?92
	8-@60	8-@60	8-@60	8-@82	8-@82	8-@82	8-A49	8-A49	8-A49	8-A49	8-A79	8-A79	8-A79	8-A79
VVSET	7-377#	8-270	8-270	8-270	8-270	8-270	8-270	8-292	8-292	8-292	8-292	8-292	8-292	8-342
	8-342	8-342	8-342	8-342	8-360	8-360	8-360	8-360	8-360	8-360	8-362#	8-387	8-387	8-387
	8-387	8-387	8-400	8-400	8-400	8-400	8-400	8-400	8-400	8-446	8-446	8-446	8-446	8-446



8-446 8-446 8-446 8-469 8-469 8-469 8-469<sup>K</sup> 2 8-469 8-469 8-469 8-469 8-469 8-526 8-526  
SEQ 0229













RELEAS 4-211# 8-342 8-360 8-446 8-446 8-469 8-469<sup>B</sup> 3 8-526 8-550 8-602 8-625 8-650 8-666 8-702  
SEQ 0233

