

RM02/03/05

RM05/3/2 DU POR TST 1
CZRMRBO

AH-F937B-MC
FICHE 1 OF 2

AUG 1981
COPYRIGHT © 80-81
MADE IN USA



The main body of the document is a large grid of approximately 15 columns and 25 rows of small, illegible text. Each cell in the grid appears to contain a small table or data entry, but the text is too faint to be read. The overall appearance is that of a microfiche or a high-density data storage format.

RM02/03/05

RM05/3/2 DU POR TST 1
CZRMRBO

AH-F937B-MC
FICHE 2 OF 2

AUG 1981
COPYRIGHT © 80-81
MADE IN USA



.REM ^

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

IDENTIFICATION

PRODUCT CODE: AC-F936B-MC
PRODUCT NAME: CZRMRB0 RM05/3/2 DUAL PORT TEST, PT 1
PRODUCT DATE: APRIL 1981
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 UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1980,1981 DIGITAL EQUIPMENT CORPORATION

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

CONTENTS

- 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 OPERATOR ACTION
- 5. OPERATING PROCEDURES
 - 5.1 'SOFTWARE' SWITCH REGISTER
 - 5.2 OPERATIONAL SWITCH SETTINGS
 - 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
- 20K MEMORY
- KW11-L OR KW11-P CLOCK TERMINAL
- RH11 OR RH70
- 1 - DISK DRIVE (RM05, RM03 OR RM02)
- RM DUAL PORT TEST CABLE (P/N: 7010507-02)

2.2 PREREQUISITE PROGRAMS

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

THE PRELIMINARY PROGRAMS MUST BE RUN TWICE: ONCE FROM EACH PORT (A & B).

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 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 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.

5. OPERATING PROCEDURES

5.1 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR 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 RM80 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR - NNNNNN NEW ='

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

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 1'S IN THE SWITCHES. ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

5.2 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

5.3 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.

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

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

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 RH11/RH70 TO J3 OF THE
RM05/3/2 BACK PANEL, THEN CONNECT THE TEST CABLE (P/N: 7010507-02)
FROM J2 TO J7 OF THE BACK PANEL AND TERMINATE THE PORT 'B' AT J6.

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

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

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

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 PORT OPTION, AND DOES NOT TEST THE UNLOAD COMMAND OR THE OPERATION OF THE PORT SELECT SWITCH ON THE DRIVE. (REFER TO PARAGRAPH 2.2 & 2.3)

7.3 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 25 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 2 MINUTES 25 SECONDS.

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

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 DISCREPANCY BETWEEN PORTS INDICATES A

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

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 '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.

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

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 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.

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.

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.

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 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'.
- 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.

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

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.
- C. ISSUE A MASSBUS CLEAR THROUGH THE RH CONTROLLER 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 CONTROLLER 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

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

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.
- 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.

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

- 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 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

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

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.
- 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.

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

- 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 RESET.

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

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 RMA. VERIFY THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMA.

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'.

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

- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD5.
- 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.

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.

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'.

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 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'.

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 RMD5.
- B. WRITE 1'S INTO RMERR 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 RMD5.
- B. WRITE 1'S INTO RMERR 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 RMD5.
- B. WAIT 500 MS AND READ RMD5 THROUGH PORT 'A'.

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

- 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 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 RMD5.
- B. WAIT 500 MS AND WRITE 0'B INTO RMD5 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 RMD5.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMD5 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 RMD5.
- B. SET PORT REQUEST BY WRITING 0'S INTO RMD5 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.

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

TEST 50 PORT 'A' SEIZE ACCESS TEST

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
597
598

```

;*LAST REVISION 04-APR-81
.TITLE CZMRB0 RM05/3/2 DU POR TST 1
;*COPYRIGHT (C) 1981
;*DIGITAL EQUIPMENT CORPORATION
;*COLORADO SPGS., CO. 80919
;*
;*PROGRAM BY MIKE LEAVITT
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-(5)), 18-MAR-81

```

599

```

.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

```

600
601

```

.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 PIR0 = 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 FC = %7             ;;PROGRAM COUNTER

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

```

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

.SBTTL RH/RM REGISTERS

;CONTROL AND STATUS REGISTER 1 (RMCS1)
602
603
604
605
606
607 000100 IE = 100 ;INTERRUPT ENABLE (BIT #6)
608 000200 RDY = 200 ;READY (BIT #7)
609 000400 A16 = 400 ;HIGH ORDER BUS ADDRESS BIT (BIT #8)
610 001000 A17 = 1000 ;HIGH ORDER BUS ADDRESS BIT (BIT #9)
611 002000 PSEL = 2000 ;PORT SELECT (BIT #10)
612 020000 MCPE = 20000 ;MASSBUS PARITY ERROR (BIT #13)
613 040000 TRE = 40000 ;TRANSFER ERROR (BIT #14)
614 100000 SC = 100000 ;SPECIAL CONDITION (BIT #15)
615
616 ;CONTROL AND STATUS REGISTER 2 (RMCS2)
617
618 000001 U0 = 1 ;UNIT SELECT (BIT #0)
619 000002 U1 = 2 ;UNIT SELECT (BIT #1)
620 000004 U3 = 4 ;UNIT SELECT (BIT #2)
621 000010 BAI = 10 ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
622 000020 PAT = 20 ;MASSBUS PARITY TEST (BIT #4)
623 000040 CLR = 40 ;CLEAR (BIT #5)
624 000100 IR = 100 ;INPUT READY (BIT #6)
625 000200 OR = 200 ;OUTPUT READY (BIT #7)
626 000400 MDPE = 400 ;MASS BUS PARITY ERROR (BIT #8)
627 001000 MXF = 1000 ;MISSED TRANSFER ERROR (BIT #9)
628 002000 PGE = 2000 ;PROGRAM ERROR (BIT #10)
629 004000 NEM = 4000 ;NON EXISTENT MEMORY (BIT #11)
630 010000 NED = 10000 ;NON EXISTENT DRIVE (BIT #12)
631 020000 UPE = 20000 ;UNIBUS PARITY ERROR (BIT #13)
632 040000 WCE = 40000 ;WRITE CHECK ERROR (BIT #14)
633 100000 DLT = 100000 ;DATA LATE (BIT #15)
634
635 ;DATA BUFFER REGISTER (RMDB)
636 ;(EACH BIT IS CALLED BY BIT NUMBER)
637
638 .SBTTL RM REGISTERS
639
    
```


RM REGISTERS

```

640          ;CONTROL AND STATUS REGISTER #1. (#00)
641
642          000001      GO      = 1          ;GO BIT (BIT #0)
643          000002      FO      = 2          ;FUNCTION CODE BIT #1
644          000004      F1      = 4          ;FUNCTION CODE BIT #2
645          000010      F2      = 10         ;FUNCTION CODE BIT #3
646          000020      F3      = 20         ;FUNCTION CODE BIT #4
647          000040      F4      = 40         ;FUNCTION CODE BIT #5
648          004000      DVA     = 4000        ;DEVICE AVAILABLE (BIT #11)
649
650          ;CONTROL STATUS REGISTER #2 (RMCS2)
651
652          000040      CLR     = BIT5         ;CONTROLLER CLEAR
653
654          ;DRIVE STATUS REGISTER (RMDS) (#01)
655
656          000001      OM      = BIT00        ;OFFSET MODE
657          000100      VV      = 100         ;VOLUME VALID (BIT #6)
658          000200      DRY     = 200         ;DRIVE READY (BIT #7)
659          000400      DPR     = 400         ;DRIVE PRESENT (BIT #8)
660          001000      PGM     = 1000        ;PROGRAMABLE (BIT #9)
661          002000      LBT     = 2000        ;LAST SECTOR TRANSFERRED (BIT #10)
662          004000      WRL     = 4000        ;WRITE LOCK (BIT #11)
663          010000      MOL     = 10000       ;MEDIUM ON-LINE (BIT #12)
664          020000      PIP     = 20000      ;POSITIONING OPERATION IN PROGRESS (BIT #13)
665          040000      ERR     = 40000      ;COMPOSITE ERROR (BIT #14)
666          100000      ATA     = 100000     ;ATTENTION ACTIVE (BIT #15)
667
668          ;ERROR REGISTER #01 (RMER1) (#02)
669
670          000001      ILF     = 1          ;ILLEGAL FUNCTION (BIT #0)
671          000002      ILR     = 2          ;ILLEGAL REGISTER (BIT #1)
672          000004      RMR     = 4          ;REGISTER MODIFICATION REFUSED (BIT #2)
673          000010      PAR     = 10         ;PARITY ERROR (BIT #3)
674          000020      FER     = 20         ;FORMAT ERROR (BIT #4)
675          000040      WCF     = 40         ;WRITE CLOCK FAIL (BIT #5)
676          000100      ECH     = 100        ;ECC HARD ERROR (BIT #6)
677          000200      HCE     = 200        ;HEADER COMPARE ERROR (BIT #7)
678          000400      HCRC    = 400        ;HEADER CRC ERROR (BIT #8)
679          001000      AOE     = 1000       ;ADDRESS OVERFLOW ERROR (BIT #9)
680          002000      IAE     = 2000      ;INVALID ADDRESS ERROR (BIT #10)
681          004000      WLE     = 4000      ;WRITE LOCK ERROR (BIT #11)
682          010000      DTE     = 10000     ;DRIVE TIMING ERROR (BIT #12)
683          020000      OPI     = 20000     ;OPERATION INCOMPLETE (BIT #13)
684          040000      UNS     = 40000     ;DRIVE UNSAFE (BIT #14)
685          100000      DCK     = 100000    ;DATA CHECK ERROR (BIT 15)
686
687          ;MAINTAINABILITY REGISTER (RMMR1) (#03)
688
689          000001      DMD     = 1          ;DIAGINOSTIC MODE (BIT #0)
690          001000      MUR     = BIT09      ;MAINTENANCE UNIT READY
691          040000      RQB     = BIT14      ;PORT B REQUEST FLOP
692          100000      RQA     = BIT15      ;PORT A REQUEST FLOP
693
694          ;ATTENTION SUMMARY PSEUDO-REGISTER (RMAS) (#04)
695
696          000001      AT0     = 1          ;DEVICE 0 (BIT #0)

```

```

697      000002      AT1      = 2      ;DEVICE 1 (BIT #1)
698      000004      AT2      = 4      ;DEVICE 2 (BIT #2)
699      000010      AT3      = 10     ;DEVICE 3 (BIT #3)
700      000020      AT4      = 20     ;DEVICE 4 (BIT #4)
701      000040      AT5      = 40     ;DEVICE 5 (BIT #5)
702      000100      AT6      = 100    ;DEVICE 6 (BIT #6)
703      000200      AT7      = 200    ;DEVICE 7 (BIT #7)
704
705      ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RMDA) (#05)
706      ;(EACH BIT IS CALLED BY BIT NUMBER)
707
708      ;DRIVE TYPE REGISTER (RMDT) (#06)
709
710      000001      DT00     = 1      ;DRIVE TYPE NUMBER BIT 1
711      000002      DT01     = 2      ;DRIVE TYPE NUMBER BIT 2
712      000004      DT02     = 4      ;DRIVE TYPE NUMBER BIT 3
713      000010      DT03     = 10     ;DRIVE TYPE NUMBER BIT 4
714      000020      DT04     = 20     ;DRIVE TYPE NUMBER BIT 5
715      000040      DT05     = 40     ;DRIVE TYPE NUMBER BIT 6
716      000100      DT06     = 100    ;DRIVE TYPE NUMBER BIT 7
717      000200      DT07     = 200    ;DRIVE TYPE NUMBER BIT 8
718      000400      DT08     = 400    ;DRIVE TYPE NUMBER BIT 9
719      004000      DRQ      = 4000   ;DRIVE REQUEST REQUIRED (BIT #11)
720      020000      MOH      = 20000  ;MOVING HEAD (BIT #13)
721      040000      TAP      = 40000  ;TAPE DRIVE (BIT #14)
722      100000      NBA      = 100000 ;NOT BLOCK ADDRESSED (BIT #15)
723
724      ;LOOK-AHEAD REGISTER (RMLA) (#07)
725
726      000100      SC0      = 100    ;SECTOR COUNT FIELD 0 (BIT #6)
727      000200      SC1      = 200    ;SECTOR COUNT FIELD 1 (BIT #7)
728      000400      SC2      = 400    ;SECTOR COUNT FIELD 2 (BIT #8)
729      001000      SC3      = 1000   ;SECTOR COUNT FIELD 3 (BIT #9)
730      002000      SC4      = 2000   ;SECTOR COUNT FIELD 4 (BIT #10)
731
732      ;RM ERROR REGISTER #2 (RMER2) (#10)
733
734      000010      DPE      = 10     ;DATA PARITY ERROR (BIT #3)
735      000200      DVC      = 200    ;DEVICE CHECK (BIT #7)
736      002000      LBC      = 2000   ;LOSS OF BIT CLOCK (BIT #10)
737      004000      LSC      = 4000   ;LOSS OF SYSTEM CLOCK (BIT #11)
738      010000      IVC      = 10000  ;INVALID COMMAND (BIT #12)
739      020000      OPE      = 20000  ;OPERATOR ERROR (BIT #13)
740      100000      SKI      = 100000 ;SEEK INCOMPLETE (BIT #14)
741
742      ;OFFSET REGISTER (RMOF) (#11)
743
744      000200      OFD      = 200    ;OFFSET FORWARD (BIT #5)
745      002000      HCI      = 2000   ;HEADER COMPARE INHIBIT (BIT #10)
746      004000      ECI      = 4000   ;ERROR CORRECTION CODE INHIBIT (BIT #11)
747      010000      FMT16    = 10000  ;FORMAT BIT (BIT #12)
748
749      ;DESIRED CYLINDER ADDRESS (RMDC) (#12)
750      ;(EACH BIT IS CALLED BY BIT NUMBER)
751
752      ;SERIAL NUMBER REGISTER (RMSN) (#14)
753      ;(EACH IS CALLED BY BIT NUMBER)

```

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

;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)

1

.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
 000176 000000

 .=174
 DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
 SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER

.SBTTL STARTING ADDRESS(ES)

2 000200 000137 002240

 JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

3 000204 000137 002246

 JMP @#START1 ;START AND CHANGE THE RH/RM ADDRESS

4
 5

.SBTTL ACT11 HOOKS

::*****
 ;HOOKS REQUIRED BY ACT11

000210
 000046
 000046 066110
 000052 000052
 000052 020000
 000210

 \$SVPC=. ;SAVE PC
 .=46
 \$ENDAD ;;1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .\$EOP
 .=52
 .WORD 20000 ;;2)SET LOC.52 TO 20000
 .=\$SVPC ;;RESTORE PC

6

0

.SBTTL COMMON TAGS

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

001100	001100			.=1100						
001100	000000			\$CMTAG:	.WORD	0			::	START OF COMMON TAGS
001102	000			\$PASS:	.WORD	0			::	CONTAINS PASS COUNT
001103	000			\$TSTNM:	.BYTE	0			::	CONTAINS THE TEST NUMBER
001104	000000			\$ERFLG:	.BYTE	0			::	CONTAINS ERROR FLAG
001106	000000			\$ICNT:	.WORD	0			::	CONTAINS SUBTEST ITERATION COUNT
001110	000000			\$LPADR:	.WORD	0			::	CONTAINS SCOPE LOOP ADDRESS
001112	000000			\$LPERR:	.WORD	0			::	CONTAINS SCOPE RETURN FOR ERRORS
001114	000			\$ERTTL:	.WORD	0			::	CONTAINS TOTAL ERRORS DETECTED
001115	001			\$ITEMB:	.BYTE	0			::	CONTAINS ITEM CONTROL BYTE
001116	000000			\$ERMAX:	.BYTE	1			::	CONTAINS MAX. ERRORS PER TEST
001120	000000			\$ERRPC:	.WORD	0			::	CONTAINS PC OF LAST ERROR INSTRUCTION
001122	000000			\$GDADR:	.WORD	0			::	CONTAINS ADDRESS OF 'GOOD' DATA
001124	000000			\$BDADR:	.WORD	0			::	CONTAINS ADDRESS OF 'BAD' DATA
001126	000000			\$GDDAT:	.WORD	0			::	CONTAINS 'GOOD' DATA
001130	000000			\$BDDAT:	.WORD	0			::	CONTAINS 'BAD' DATA
001132	000000				.WORD	0			::	RESERVED--NOT TO BE USED
001134	000			\$AUTOB:	.BYTE	0			::	AUTOMATIC MODE INDICATOR
001135	000			\$INTAG:	.BYTE	0			::	INTERRUPT MODE INDICATOR
001136	000000				.WORD	0				
001140	177570			\$SWR:	.WORD	DSWR			::	ADDRESS OF SWITCH REGISTER
001142	177570			\$DISPLAY:	.WORD	DDISP			::	ADDRESS OF DISPLAY REGISTER
001144	177560			\$TKS:	177560				::	TTY KBD STATUS
001146	177562			\$TKB:	177562				::	TTY KBD BUFFER
001150	177564			\$TPS:	177564				::	TTY PRINTER STATUS REG. ADDRESS
001152	177566			\$TPB:	177566				::	TTY PRINTER BUFFER REG. ADDRESS
001154	000			\$NULL:	.BYTE	0			::	CONTAINS NULL CHARACTER FOR FILLS
001155	002			\$FILLS:	.BYTE	2			::	CONTAINS # OF FILLER CHARACTERS REQUIRED
001156	012			\$FILLC:	.BYTE	12			::	INSERT FILL CHARS. AFTER A 'LINE FEED'
001157	000			\$TPFLG:	.BYTE	0			::	'TERMINAL AVAILABLE' FLAG (BIT<07>-0=YES)
001160	000000			\$REGAD:	.WORD	0			::	CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED
001162	000000			\$REGO:	.WORD	0			::	CONTAINS ((\$REGAD)+0)
001164	000000			\$TMP0:	.WORD	0			::	USER DEFINED
001166	000000			\$TMP1:	.WORD	0			::	USER DEFINED
001170	000000			\$TMP2:	.WORD	0			::	USER DEFINED
001172	000000			\$TMP3:	.WORD	0			::	USER DEFINED
001174	000000			\$TMP4:	.WORD	0			::	USER DEFINED
001176	000000			\$TIMES:	0				::	MAX. NUMBER OF ITERATIONS
001200	000000			\$ESCAPE:	0				::	ESCAPE ON ERROR ADDRESS
001202	207	377	377	\$BELL:	.ASCIZ	<207><377><377>			::	CODE FOR BELL
001206	077			\$QUES:	.ASCII	/?/			::	QUESTION MARK
001207	015			\$CRLF:	.ASCII	<15>			::	CARRIAGE RETURN
001210	012	000		\$LF:	.ASCIZ	<12>			::	LINE FEED

.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

.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				
2					
3					
4	001310	072724	EM1		;WRONG DRIVE TYPE
5	001312	077341	DH1		
6	001314	101230	DT1		
7	001316	101516	DF1		
8					
9					
10					
11	001320	072745	EM2		;DRIVE NOT ON LINE
12	001322	077341	DH1		
13	001324	101230	DT1		
14	001326	101516	DF1		
15					
16					
17					
18	001330	072767	EM3		;SERIAL NUMBERS NOT THE SAME
19	001332	077412	DH3		
20	001334	101244	DT3		
21	001336	101516	DF1		
22					
23					
24					
25	001340	073051	EM4		;DRIVE NOT SEIZED BY PORT 'N'
26	001342	077461	DH4		
27	001344	101312	DT7		
28	001346	101531	DF7		
29					
30					
31					
32	001350	073102	EM5		;WRONG STATUS SEEN BY THE SEIZING PORT
33	001352	077604	DH5		
34	001354	101260	DT5		
35	001356	101523	DF5		
36					
37					
38					
39	001360	073150	EM6		;REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED
40	001362	100054	DH13		
41	001364	101332	DT13		
42	001366	101523	DF5		

43				
44			:ERROR 7	
45				
46	001370	073250	EM7	:REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
47	001372	077660	DH7	
48	001374	101312	DT7	
49	001376	101531	DF7	
50				
51			:ERROR 10	
52				
53	001400	073331	EM10	:REGISTER CONTENTS INCORRECT
54	001402	077604	DH5	
55	001404	101260	DT5	
56	001406	101523	DF5	
57				
58			:ERROR 11	
59				
60	001410	073361	EM11	:CONTROL BUS PARITY ERROR WHILE READING REGISTER
61	001412	100003	DH11	
62	001414	101230	DT1	
63	001416	101516	DF1	
64				
65			:ERROR 12	
66				
67	001420	073445	EM12	:DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
68	001422	100550	DH36	
69	001424	101420	DT37	
70	001426	101544	DF36	
71				
72			:ERROR 13	
73				
74	001430	073515	EM13	: 'VOLUME VALID' BIT NOT SET BY READ IN PRESET
75	001432	100054	DH13	
76	001434	101332	DT13	
77	001436	101523	DF5	
78				
79			:ERROR 14	
80				
81	001440	073602	EM14	: 'VOLUME VALID' SET ON THE OPPOSITE PORT
82	001442	100054	DH13	
83	001444	101332	DT13	
84	001446	101523	DF5	
85				
86			:ERROR 15	
87				
88	001450	073645	EM15	:THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
89	001452	077660	DH7	
90	001454	101312	DT7	
91	001456	101531	DF7	
92				
93			:ERROR 16	
94				
95	001460	073724	EM16	:ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
96	001462	077660	DH7	
97	001464	101312	DT7	
98	001466	101531	DF7	
99				

100			:ERROR 17	
101				
102	001470	073777	EM17	:ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
103	001472	077660	DH7	
104	001474	101312	DT7	
105	001476	101531	DF7	
106				
107			:ERROR 20	
108				
109	001500	074056	EM20	:DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
110	001502	100550	DH36	
111	001504	101420	DT37	
112	001506	101544	DF36	
113				
114			:ERROR 21	
115				
116	001510	074136	EM21	:DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
117	001512	100550	DH36	
118	001514	101420	DT37	
119	001516	101544	DF36	
120				
121			:ERROR 22	
122				
123	001520	074211	EM22	:DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
124	001522	100174	DH22	
125	001524	101350	DT22	
126	001526	101540	DF31	
127				
128			:ERROR 23	
129				
130	001530	074276	EM23	:TIMEOUT CLEARED THE DRIVE'S ERROR BIT
131	001532	100272	DH23	
132	001534	101362	DT23	
133	001536	101516	DF1	
134				
135			:ERROR 24	
136				
137	001540	074344	EM24	:RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
138	001542	100272	DH23	
139	001544	101362	DT23	
140	001546	101516	DF1	
141				
142				
143			:ERROR 25	
144				
145	001550	074423	EM25	:TIMEOUT ONE-SHOT DID NOT RETRIGGER
146	001552	100550	DH36	
147	001554	101410	DT36	
148	001556	101544	DF36	
149				
150				
151			:ERROR 26	
152				
153	001560	074466	EM26	:DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
154	001562	100174	DH22	
155	001564	101350	DT22	
156	001566	101540	DF31	

157				
158			:ERROR 27	
159				
160	001570	074553	EM27	:REGISTER WRONG AFTER RELEASE WITH REQUEST SET
161	001572	077660	DH7	
162	001574	101312	DT7	
163	001576	101531	DF7	
164				
165			:ERROR 30	
166				
167	001600	074631	EM30	:DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
168	001602	100550	DH36	
169	001604	101410	DT36	
170	001606	101544	DF36	
171				
172			:ERROR 31	
173				
174	001610	074726	EM31	:DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SET
175	001612	100451	DH31	
176	001614	101376	DT31	
177	001616	101540	DF31	
178				
179			:ERROR 32	
180				
181	001620	075003	EM32	:ATTN BIT WRONG AFTER RECALIBRATE COMMAND
182	001622	077604	DH5	
183	001624	101260	DT5	
184	001626	101523	DF5	
185				
186			:ERROR 33	
187				
188	001630	075054	EM33	:DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED
189	001632	100550	DH36	
190	001634	101410	DT36	
191	001636	101544	DF36	
192				
193			:ERROR 34	
194				
195	001640	075156	EM34	:DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED
196	001642	100550	DH36	
197	001644	101410	DT36	
198	001646	101544	DF36	
199				
200			:ERROR 35	
201				
202	001650	075261	EM35	:DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT ONE SHOT
203	001652	100550	DH36	
204	001654	101420	DT37	
205	001656	101544	DF36	
206				
207			:ERROR 36	
208				
209	001660	075340	EM36	:TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
210	001662	100550	DH36	
211	001664	101410	DT36	
212	001666	101544	DF36	
213				

214			:ERROR 37	
215				
216	001670	075412	EM37	:DRIVE IS NON-EXISTENT
217	001672	100550	DH36	
218	001674	101420	DT37	
219	001676	101544	DF36	
220				
221			:ERROR 40	
222				
223	001700	075460	EM40	:ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
224	001702	077341	DH1	
225	001704	101362	DT23	
226	001706	101516	DF1	
227				
228			:ERROR 41	
229				
230	001710	075535	EM41	:TIMEOUT CLEARED ATTENTION BIT
231	001712	100272	DH23	
232	001714	101362	DT23	
233	001716	101516	DF1	
234				
235			:ERROR 42	
236				
237	001720	075577	EM42	:DRIVE NOT IN NEUTRAL OR SEIZED
238	001722	100577	DH42	
239	001724	101430	DT42	
240	001726	101547	DF42	
241				
242			:ERROR 43	
243				
244	001730	075665	EM43	:DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
245	001732	100577	DH42	
246	001734	101430	DT42	
247	001736	101547	DF42	
248				
249			:ERROR 44	
250				
251	001740	075742	EM44	:WRITE ATTENTION BIT DID NOT SET PORT REQUEST
252	001742	100616	DH44	
253	001744	101376	DT31	
254	001746	101540	DF31	
255				
256			:ERROR 45	
257				
258	001750	076017	EM45	:PORT SELECT SWITCH ON DRIVE NOT IN 'A/B'
259	001752	077341	DH1	
260	001754	101230	DT1	
261	001756	101516	DF1	
262				
263			:ERROR 46	
264				
265	001760	076071	EM46	:CAN'T ACCESS DRIVE THROUGH EITHER PORT
266	001762	100714	DH46	
267	001764	101436	DT46	
268	001766	101540	DF31	
269				
270			:ERROR 47	

272	001770	076140	EM47	:ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
273	001772	100272	DH23	
274	001774	101362	DT23	
275	001776	101516	DF1	
276				
277				:ERROR 50
278				
279	002000	076226	EM50	:ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMAND
280	002002	100054	DH13	
281	002004	101332	DT13	
282	002006	101523	DF5	
283				
284				:ERROR 51
285				
286	002010	076310	EM51	:ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
287	002012	077604	DH5	
288	002014	101260	DT5	
289	002016	101523	DF5	
290				
291				:ERROR 52
292				
293	002020	076377	EM52	:ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
294	002022	100054	DH13	
295	002024	101332	DT13	
296	002026	101523	DF5	
297				
298				:ERROR 53
299				
300	002030	076472	EM53	:CAN'T READ ATTN BIT FROM OPPOSITE PORT
301	002032	100272	DH23	
302	002034	101230	DT1	
303	002036	101516	DF1	
304				
305				:ERROR 54
306				
307	002040	076553	EM54	:RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
308	002042	100174	DH22	
309	002044	101450	DT54	
310	002046	101540	DF31	
311				
312				:ERROR 55
313				
314	002050	076646	EM55	:TIMEOUT ONE-SHOT IS LESS THAN 500 MS
315	002052	101011	DH55	
316	002054	101462	DT55	
317	002056	101551	DF55	
318				
319				:ERROR 56
320				
321	002060	076713	EM56	:RH/RM DIDN'T RESPOND TO ADDRESSING
322	002062	101067	DH56	
323	002064	101474	DT56	
324	002066	101555	DF56	
325				
326				
327				:ERROR 57

ERROR POINTER TABLE

328				
329	002070	076756	EM57	;PORT REQUEST FLOPS WRONG
330	002072	101076	DH57	
331	002074	101500	DT57	
332	002076	101523	DF5	
333				
334			.ERROR	60
335				
336	002100	077017	EM60	;ATTENTION BITS NOT RESET BY RMAS
337	002102	077604	DH5	
338	002104	101260	DT5	
339	002106	101523	DF5	
340				
341			:ERROR	61
342				
343	002110	077063	EM61	;ATTENTION NOT RESET BY GO
344	002112	100272	DH23	
345	002114	101362	DT23	
346	002116	101516	DF1	
347				
348			:ERROR	62
349				
350	002120	077115	EM62	;ATTENTION RESET BY GO WHEN NOT SEIZED
351	002122	100054	DH13	
352	002124	101332	DT13	
353	002126	101523	DF5	
354				
355			:ERROR	63
356				
357	002130	077163	EM63	;DRIVE SEIZED BY UNIT READY CHANGE
358	002132	100550	DH36	
359	002134	101410	DT36	
360	002136	101544	DF36	
361				
362			:ERROR	64
363				
364	002140	077225	EM64	;ATTENTION NOT SET BY UNIT READY CHANGE
365	002142	077660	DH7	
366	002144	101312	DT7	
367	002146	101531	DF7	
368				
369			:ERROR	65
370				
371	002150	077274	EM65	;VV NOT RESET BY UNIT READY
372	002152	077604	DH5	
373	002154	101260	DT5	
374	002156	101523	DF5	
375				

```

1      ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
2
3 002160 011600 BAD1MO: MOV (SP),R0 ;SAVE PC WHERE THE TIME OUT OCCURED
4 002162 005740 TST -(R0) ;ADJUST PC -2
5 002164 022626 CMP (SP)+,(SP)+ ;RESTORE STACK POINTER
6 002166 104401 002174 TYPE ,65$ ;:TYPE ASCIZ STRING
   002172 000417 BR 64$ ;:GET OVER THE ASCIZ
   ;:65$: .ASCIZ <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
   64$:
7 002232 010046 MOV R0,-(SP) ;SETUP FOR TYPING OUT PC
8 002234 104402 TYPOC
9 002236 000240 NOP ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
   ;TO STOP ON UNEXPECTED TIMEOUT.
10
11
12 .SBTTL START OF PROGRAM
13
14 002240 005037 001302 START: CLR CHGADR ;CLEAR THE 'CHANGE RH/RM ADDRESS' INDICATOR
15 002244 000403 BR START2 ;GO TO THE START
16
17 002246 012737 177777 001302 START1: MOV #-1,CHGADR ;SET THE 'CHANGE RH/RM ADDRESS' INDICATOR
18
19 002254 000240 START2: NOP
20 002256 005227 000000 INC #0 ;TTY LOOP, WAIT FOR INCREMENT
21 002262 001375 BNE .-4 ;OF WORD
22 002264 000005 RESET ;CLEAR THE WORLD
23
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 #,$SLPADR ;:INITIALIZE THE LOOP ADDRESS FOR SCOPE
   MOV #,$SLPERR ;: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
   BK 65$ ;:BRANCH IF NO TIMEOUT
   64$: MOV #65$, (SP) ;:SET UP FOR TRAP RETURN
    
```



```

002456 000002 RTI
002460 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
002466 012737 000174 001142 MOV #DISPREG,DISPLAY
002474 012637 000004 66$: 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
29 .SETTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
002514 005227 177777 INC #-1 ;;FIRST TIME?
002520 001037 BNE 67$ ;;BRANCH IF NO
002522 022737 066110 000042 CMP #SENDAD,@#42 ;;ACT-11?
002530 001433 BEQ 67$ ;;BRANCH IF YES
002532 104401 002540 TYPE ,68$ ;;TYPE ASCIZ STRING
002536 000430 BR 67$ ;;GET OVER THE ASCIZ
;;68$: .ASCIZ <CRLF>@CZMRBO - RM05/3/2 DUAL PORT LOGIC TEST, PT 1@<CRLF>
67$:
.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
002620 005737 000042 TST @#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
002624 001006 BNE 69$ ;;BRANCH IF YES
002626 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
002634 001005 BNE 70$ ;;BRANCH IF NO
002636 104406 GTSWR ;;GET SOFT-SWR SETTINGS
002640 000403 BR 70$
002642 112737 000001 001134 69$: MOV #1,$AUTOB ;;SET AUTO-MODE INDICATOR
002650 70$:

30
31 002650 004737 070610 JSR PC,$TKINT ;SETUP THE TTY KEYBOARD
32 002654 004737 003246 1$: JSR PC,CHANGE ;CHECK/CHANGE THE RH/RM ADDRESS
33 002660 104401 072372 TYPE ,ENTERA ;ENTER DRIVE ADDRESS
34 002664 104412 RDOCT ;GET THE ADDRESS
35 002666 012637 001224 MOV (SP)+,PORTA ;STORE THE ADDRESS
36 002672 023727 001224 000007 CMP PORTA,#7 ;SEE IF ADDRESS TOO LARGE
37 002700 101403 BLOS 2$ ;BR IF NOT
38 002702 104401 072421 TYPE ,ADRERR ;TYPE ADDRESS ERROR MESSAGE
39 002706 000762 BR 1$ ;TRY AGAIN
40 002710 013737 001224 001226 2$: MOV PORTA,PORTB ;GENERATE THE PORT B ADDRESS
41 002716 005237 001226 INC PORTB ;INCREMENT THE ADDRESS
42 002722 042737 000016 001226 BIC #16,PORTB ;LEAVE BIT 0
43 002730 013746 001224 MOV PORTA,-(SP) ;PUT PORT A ADDRESS ON THE STACK
44 002734 042716 177771 BIC #^C6,(SP) ;SAVE BITS 1 & 2
45 002740 052637 001226 BIS (SP)+,PORTB ;SET BITS 1 & 2 IN PORT B ADDRESS
46 002744 104401 072444 TYPE ,PORTAIS ;'PORT A ADDRESS IS '
47 002750 013746 001224 MOV PORTA,-(SP) ;SAVE PORTA FOR TYPEOUT
;;TYPE PORT A ADDRESS
;;GO TYPE--OCTAL ASCII
002754 104403 TYPOS ;TYPE PORT A ADDRESS
002756 001 .BYTE 1 ;TYPE 1 DIGIT(S)
002757 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
48 002760 104401 072473 TYPE ,PORTBIS ;'PORT B ADDRESS IS '
49 002764 013746 001226 MOV PORTB,-(SP) ;SAVE PORTB FOR TYPEOUT
;;TYPE PORT B ADDRESS
;;GO TYPE--OCTAL ASCII
002770 104403 TYPOS ;TYPE PORT B ADDRESS
002772 001 .BYTE 1 ;TYPE 1 DIGIT(S)
002773 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
50 002774 104401 001207 TYPE ,$CRLF ;ANOTHER CR-LF

```

```

51 003000 013737 001224 001230      MOV    PORTA,PORTC      ;GENERATE ADDRESS OF DRIVE NOT TESTED
52 003006 062737 000006 001230      ADD    #6,PORTC        ;COMPLEMENT SOME BITS
53 003014 042737 177770 001230      BIC    #^C7,PORTC     ;SAVE ONLY LOWER BITS
54 003022 013701 001224                MOV    PORTA,R1        ;USE PORT A ADDRESS AS INDEX
55 003026 116137 101672 001236      MOVB   ATABIT(R1),ASR1 ;GET ATTENTION BIT FOR DRIVE
58 003034 005037 001262                CLR    TIMEA          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003040 005037 001264                CLR    TIMEAP         ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003044 005037 001270                CLR    TIMEB          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003050 005037 001272                CLR    TIMEBP         ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
59 003054 004737 066130      JSR    PC,CKCLK        ;SETUP CLOCK
60 003060 000137 003074      JMP    EXEC            ;CLOCK HAS BEEN STARTED
61 003064 104401 072522                TYPE   ,NOCLOCK       ;NO CLOCK ON SYSTEM
62 003070 000000                3$:   HALT            ;FATAL ERROR
63 003072 000776                BR     3$             ;INTERLOCK THE HALT
64
65                                     ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
66
67 003074 000005      EXEC:  RESET          ;CLEAR EVERYTHING
68 003076 005037 177776      CLR    PS             ;CLEAR THE PROCESSOR STATUS WORD
69 003102 104401 001207      TYPE   ,$CRLF        ;CR-LF
70 003106 013700 001304      MOV    $RMADR,RO      ;RH/RM ADDRESS FOR INDEXING
71 003112 012706 001100      MOV    #STACK,SP     ;LOAD STACK POINTER
72 003116 004737 066130      JSR    PC,CKCLK       ;START THE CLOCK
73 003122 000240                NOP                    ;RETURN IF NO CLOCK
74 003124 004737 070610      JSR    PC,$TKINT     ;INITIALIZE THE KEYBOARD
75 003130 005037 001300      CLR    KYBCTL         ;CLEAR SINGLE TEST INDICATOR
76 003134 005037 001100      CLR    $PASS         ;CLEAR THE PASS COUNT
77 003140 112737 000001 001115      MOVB   #1,$ERMAX     ;SET ERROR MAX TO 1
78 003146 012737 003146 001106      MOV    #,$SLPADR     ;INITIAL SETTING FOR LOOP ADDRESS
79 003154 012737 003154 001110      MOV    #,$SLPERR     ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
80 003162 104401 072560      1$:   TYPE   ,TESTNO  ;ASK FOR TEST NUMBER
81 003166 104412                RDOCT                ;GET THE NUMBER
82 003170 012601                MOV    (SP)+,R1      ;PUT ENTRY INTO R1
83 003172 001002                BNE   2$             ;BR IF NOT ZERO
84 003174 000137 003374      JMP    TST1           ;ENTER ZERO - PERFORM ALL TESTS
85 003200 020137 101702      2$:   CMP    R1,MAXTN  ;SEE IF NUMBER GREATER THAN MAXIMUM
86 003204 003403                BLE   3$             ;BR IF LESS OR EQUAL
87 003206 104401 072600      TYPE   ,BADNO        ;BAD ENTRY
88 003212 000763                BR     1$            ;TRY AGAIN
89 003214 005301      3$:   DEC    R1        ;DECREMENT ENTRY
90 003216 006301                ASL   R1             ;SHIFT IT LEFT
91 003220 016137 101556 003244      MOV    TSTADR(R1),4$ ;GET THE TEST ADDRESS
92 003226 005237 001300                INC   KYBCTL         ;SET SINGLE TEST INDICATOR
93 003232 012737 000001 001104      MOV    #1,$ICNT     ;PRESET ITERATION COUNT
94 003240 000177 000000                JMP   @4$            ;GO TO THE SELECTED TEST
95 003244 000000      4$:   .WORD  0        ;TEST ADDRESS GOES HERE
96
97                                     ;CHANGE THE RH/RM UNIBUS ADDRESS USED BY THE PROGRAM
98
99 003246 005737 001302      CHANGE: TST    CHGADR ;CHANGE THE ADDRESS ?
100 003252 001421                BEQ   3$             ;BR IF NOT
101 003254 005037 001302      CLR    CHGADR        ;CLEAR THE INDICATOR
102 003260 104401 072640      1$:   TYPE   ,ADDRIS  ;TYPE OUT WHAT THE PRESENT ADDRESS IS
103 003264 013746 001304      MOV    $RMADR,-(SP)  ;PUT THE ADDRESS ON THE STACK
104 003270 104402                TYPOC                ;TYPE THE ACTUAL ADDRESS
105 003272 104401 001207      TYPE   ,$CRLF        ;CR-LF
106 003276 104401 072675      TYPE   ,NTRH        ;ASK FOR NEW ADDRESS
    
```

107	003302	104412				RDOCT		
108	003304	005716				TST	(SP)	:0 OR 'CR' ENTERED ?
109	003306	001402				BEQ	2\$:BR IF EITHER ENTERED (NO ADDRESS CHANGE)
110	003310	011637	001304			MOV	(SP), \$RMADR	:NEW RH/RM ADDRESS
111	003314	005726			2\$:	TST	(SP)+	:CORRECT THE STACK POINTER
112	003316	012737	003336	000004	3\$:	MOV	#4\$, @#4	:LOAD TRAP ADDRESS
113	003324	013700	001304			MOV	\$RMADR, R0	:GET RH/RM ADDRESS
114	003330	005760	000002			TST	RMWC(R0)	:RESPONDS AT THAT ADDRESS ?
115	003334	000404				BR	5\$:BR IF YES
116	003336				4\$:			
	003336	104056				EMT	56	
117	003340	062706	000004			ADD	#4, SP	:RESET THE STACK POINTER
118	003344	000745				BR	1\$:GET ADDRESS AGAIN
119	003346	012737	000006	000004	5\$:	MOV	#6, @#4	:RESTORE THE VECTOR
120	003354	000207				RTS	PC	:RETURN

TESTS

1
2
16
17
18
19
20

.SBTL TESTS

003356 013700 001304
003362 012746 000240
003366 012746 003374
003372 000002
003374

TST1AA: MOV \$RMADR,R0 ;:RESTORE R0 AFTER END OF PASS
MOV #PRS,-(SP) ;:PUT NEW PS ON STACK
MOV #64\$,-(SP) ;:PUT NEW PC ON STACK
RTI ;:POP NEW PC AND PS

64\$:

```

:*****
: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, 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.
:*****
    
```

003374
003374 005737 001300
003400 001406
003402 100002
003404 000137 003074
003410 012737 177777 001300
003416 012737 003432 001106
003424 012737 003432 001110
003432
003432 112737 000001 001102
003440 012706 001100
003444 012737 000001 001176

```

TST1:
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 #TEST1,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST1,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST1:
MOVB #1,$STSTM ;MOVE #1 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #1,$TIMES ;:DO 1 ITERATION
MOV #CLR,RMCS2(R0) ;INITIALIZE THE MASSBUS
    
```

21
22
23
24
25
33

003460 113760 001224 000010
003466 013737 001224 001240
003474 005760 000012
003500 005037 001250
003504 016037 000010 001126
003512 012737 000010 001122
003520 060037 001122
003524 005037 001124
003530 013737 001126 001164
003536 042737 167777 001164
003544 023737 001124 001164
003552 001414
003554 013737 001126 001174
003562 042737 010000 001174
003570 053737 001174 001124
003576 104037
003600 005137 001250

```

:VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B
MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
TST RMDS(R0) ;SEE IF DRIVE (PORT A) PRESENT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMCS2(R0),$BDDAT ;GET CONTENTS OF RMCS2
MOV #RMCS2,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0,$BADDR ;ADD RH/RM BASE ADDRESS
CLR $GDDAT ;WHAT REGISTER SHOULD BE
MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO 'TMP0'
BIC #^CNED,$TMP0 ;SAVE SPECIFIED BITS
CMP $GDDAT,$TMP0 ;COMPARE THE BITS
BEQ 64$ ;BR IF OK
MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
BIC #NED,$TMP4 ;CLEAR THE MASKED BITS
BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
EMT 37
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
    
```

```

003604 000240          64$:  NOP
003606 005737 001250    TS*   CKERR          ;WAS 'NED' SET ?
003612 001403          BEQ   .+10       ;BR IF NOT
003614 012760 000040 000010  MOV  #CLR, RMCS2(R0) ;ISSUE MASSBUS INIT TO CLEAR 'NED'
003622 113760 001226 000010  MOV  PORTB, RMCS2(R0) ;SELECT PORT B
003630 013737 001226 001240  MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
003636 005760 000012    TST  RMD5(R0)    ;SEE IF DRIVE (PORT B) PRESENT
003642 005037 001250    CLR  CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
003646 016037 000010 001126  MOV  RMCS2(R0), $BDDAT ;GET CONTENTS OF RMCS2
003654 012737 000010 001122  MOV  #RMCS2, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
003662 060037 001122    ADD  R0, $BDADR   ;ADD RH/RM BASE ADDRESS
003666 005037 001124    CLR  $GDDAT      ;WHAT REGISTER SHOULD BE
003672 013737 001126 001164  MOV  $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
003700 042737 167777 001164  BIC  #^CNED, $TMP0 ;SAVE SPECIFIED BITS
003706 023737 001124 001164  CMP  $GDDAT, $TMP0 ;COMPARE THE BITS
003714 001414          BEQ   66$        ;BR IF OK
003716 013737 001126 001174  MOV  $BDDAT, $TMP1  ;COPY 'BAD DATA'
003724 042737 010000 001174  BIC  #NED, $TMP4   ;CLEAR THE MASKED BITS
003732 053737 001174 001124  BIS  $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
003740 104037          FMT  37
003742 005137 001250    COM  CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
003746 000240          66$:  NOP
003750 005737 001250    TST  CKERR       ;WAS 'NED' SET ?
003754 001403          BEQ   .+10       ;BR IF NOT
003756 012760 000040 000010  MOV  #CLR, RMCS2(R0) ;ISSUE MASSBUS INIT TO CLEAR 'NED'

;CONFIRM THAT DRIVE IS AN RM05, RM03 OR RM02 AND IS DUAL PORTED

34
35
36
40 003764 113760 001224 000010  MOV  PORTA, RMCS2(R0) ;SELECT PORT A
003772 013737 001224 001240  MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004000 005037 001250    CLR  CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
004004 016037 000026 001126  MOV  RMDT(R0), $BDDAT ;GET CONTENTS OF RMDT
004012 012737 000026 001122  MOV  #RMDT, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004020 060037 001122    ADD  R0, $BDADR   ;ADD RH/RM BASE ADDRESS
004024 012737 024027 001124  MOV  #024027, $GDDAT ;WHAT REGISTER SHOULD BE
004032 022737 024024 001126  CMP  #024024, $BDDAT ;DUAL PORT RM03 ?
004040 001413          BEQ   68$        ;YES !!
004042 022737 024025 001126  CMP  #024025, $BDDAT ;DUAL PORT RM02 ?
004050 001407          BEQ   68$        ;YES !!
004052 023737 001124 001126  CMP  $GDDAT, $BDDAT ;IS THE REGISTER OK ?
004060 001403          BEQ   68$        ;BR IF OK
004062 104001          EMT  1
004064 005137 001250    COM  CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
004070 000240          68$:  NOP
004072 113760 001226 000010  MOV  PORTB, RMCS2(R0) ;SELECT PORT B
004100 013737 001226 001240  MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004106 005037 001250    CLR  CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
004112 016037 000026 001126  MOV  RMDT(R0), $BDDAT ;GET CONTENTS OF RMDT
004120 012737 000026 001122  MOV  #RMDT, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004126 060037 001122    ADD  R0, $BDADR   ;ADD RH/RM BASE ADDRESS
004132 012737 024027 001124  MOV  #024027, $GDDAT ;WHAT REGISTER SHOULD BE
004140 022737 024024 001126  CMP  #024024, $BDDAT ;DUAL PORT RM03 ?
004146 001413          BEQ   70$        ;YES !!
004150 022737 024025 001126  CMP  #024025, $BDDAT ;DUAL PORT RM02 ?
004156 001407          BEQ   70$        ;YES !!
004160 023737 001124 001126  CMP  $GDDAT, $BDDAT ;IS THE REGISTER OK ?
004166 001403          BEQ   70$        ;BR IF OK

```

```

004170 104001          EMT      1
004172 005137 001250  COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
004176 000240          NOP
70$:
;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
41
42
43
48 004200 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
004206 013737 001224 001240  MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004214 005037 001250          CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
004220 016037 000012 001126  MOV    RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
004226 012737 000012 001122  MOV    #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004234 060037 001122          ADD    R0, $BDADR ;ADD RH/RM BASE ADDRESS
004240 012737 001000 001124  MOV    #PGM, $GDDAT ;WHAT REGISTER SHOULD BE
004246 013737 001126 001164  MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
004254 042737 176777 001164  BIC    #^CPGM, $TMP0 ;SAVE SPECIFIED BITS
004262 023737 001124 001164  CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
004270 001414          BEQ    72$ ;BR IF OK
004272 013737 001126 001174  MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
004300 042737 001000 001174  BIC    #PGM, $TMP4 ;CLEAR THE MASKED BITS
004306 053737 001174 001124  BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004314 104045          EMT      45
004316 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
004322 000240          NOP
72$:
004324 005037 001250          CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
004330 016037 000012 001126  MOV    RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
004336 012737 000012 001122  MOV    #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004344 060037 001122          ADD    R0, $BDADR ;ADD RH/RM BASE ADDRESS
004350 012737 010600 001124  MOV    #MOL!DPR!DRY, $GDDAT ;WHAT REGISTER SHOULD BE
004356 013737 001126 001164  MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
004364 042737 167177 001164  BIC    #^C10600, $TMP0 ;SAVE SPECIFIED BITS
004372 023737 001124 001164  CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
004400 001414          BEQ    74$ ;BR IF OK
004402 013737 001126 001174  MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
004410 042737 010600 001174  BIC    #10600, $TMP4 ;CLEAR THE MASKED BITS
004416 053737 001174 001124  BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004424 104002          EMT      2
004426 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
004432 000240          NOP
74$:
004434 113760 001226 000010  MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
004442 013737 001226 001240  MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
004450 005037 001250          CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
004454 016037 000012 001126  MOV    RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
004462 012737 000012 001122  MOV    #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004470 060037 001122          ADD    R0, $BDADR ;ADD RH/RM BASE ADDRESS
004474 012737 001000 001124  MOV    #PGM, $GDDAT ;WHAT REGISTER SHOULD BE
004502 013737 001126 001164  MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
004510 042737 176777 001164  BIC    #^CPGM, $TMP0 ;SAVE SPECIFIED BITS
004516 023737 001124 001164  CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
004524 001414          BEQ    76$ ;BR IF OK
004526 013737 001126 001174  MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
004534 042737 001000 001174  BIC    #PGM, $TMP4 ;CLEAR THE MASKED BITS
004542 053737 001174 001124  BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
004550 104045          EMT      45
004552 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
004556 000240          NOP
76$:
004560 005037 001250          CLR    CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
004564 016037 000012 001126  MOV    RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5

```



```

004572 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004600 060037 001122      ADD      R0,$BDADR   ;ADD RH/RM BASE ADDRESS
004604 012737 010600 001124      MOV      #MOL!DPR.DRY,$GDDAT ;WHAT REGISTER SHOULD BE
004612 013737 001126 001164      MOV      $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
004620 042737 167177 001164      BIC      #^C10600,$TMP0 ;SAVE SPECIFIED BITS
004626 023737 001124 001164      CMP      $GDDAT,$TMP0   ;COMPARE THE BITS
004634 001414      BEQ      78$           ;BR IF OK
004636 013737 001126 001174      MOV      $BDDAT,$TMP4   ;COPY 'BAD DATA'
004644 042737 010600 001174      BIC      #10600,$TMP4   ;CLEAR THE MASKED BITS
004652 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
004660 104002      EMT      2
004662 005137 001250      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
004666 000240      78$:     NOP
    
```

49
 50
 51
 52 :VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME

```

52 004670 113760 001226 000010      MOVVB   PORTA,RMCS2(R0) ;SELECT PORT A
53 004676 016037 000030 001124      MOV      RMSN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
54 004704 113760 001226 000010      MOVVB   PORTB,RMCS2(R0) ;SELECT PORT B
55 004712 016037 000030 001126      MOV      RMSN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
56 004720 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;ARE THEY THE SAME ?
57 004726 001406      BEQ      1$           ;BR IF THEY ARE
58 004730 104003      EMT      3
59 004732 032777 100000 174200      BIT      #SW15,@SWR     ;HALT ON ERROR ?
60 004740 001001      BNE      1$           ;BR IF SET - PROGRAM HAS ALREADY HALTED
61 004742 000000      HALT
62 004744 000004      1$:     SCOPE         ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
                                ;LOOP ?
    
```

63
 81
 82
 :*****
 :*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.
 :*
 :*****

```

004746 005737 001300      TST2:   TST      #YBCTL     ;PERFORMING ONLY SINGLE TEST ?
004746 001406      BEQ      2$           ;BR IF NOT
004754 100002      BPL      1$           ;BR IF JUST ENTERED TEST
004756 000137 003074      JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
004762 012737 177777 001300      1$:     MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
004770 012737 005004 001106      2$:     MOV      #TEST2,$LPADR ;SETUP SCOPE LOOP ADDRESS
004776 012737 005004 001110      MOV      #TEST2,$LPERR ;SETUP ERROR LOOP ADDRESS
005004      TEST2:
005004 112737 000002 001102      MOVVB   #2,$STNM     ;MOVE #2 TO TEST NUMBER
    
```

```

005012 012706 001100      MOV      #STACK,SP      ;LOAD THE STACK POINTER
005016 012737 000002 001176  MOV      #2, $TIMES      ;DO 2. ITERATIONS

R3
151 005024 012737 000240 177776  MOV      #<5*32.>, @#PS  ;SET PRIORITY TO 5 IN CASE LOOPING
005032 005037 001262      CLR      TIMEA          ;CLEAR TIMEOUT VALUE FOR PORT A
005036 005037 001264      CLR      TIMEAP        ;CLEAR UPPER TIMEOUT TOLERANCE
005042 005037 001266      CLR      TIMEAM        ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

005046 005037 001256      CLR      TIME          ;CLEAR THE ELAPSED TIME COUNTER
005052 012737 003720 001260  MOV      #2000.,WATCH  ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT A

005060 113760 001224 000010  MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
005066 013737 001224 001242  MOV      PORTA, SEIZPT  ;STORE SEIZING PORT'S ADDRESS
005074 005060 000006      CLR      RMDA(R0)      ;WRITE RMDA
005100 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
005106 013737 001226 001240  MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005114 013737 001226 001244  MOV      PORTB, OPPRT   ;'OPPOSITE' PORT ADDRESS
005122 016037 000012 001126  MOV      RMD5(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
005130 010037 001122      MOV      R0, $BADDR    ;RH/RM BASE ADDRESS
005134 062737 000012 001122  ADD      #RMD5, $BADDR  ;GENERATE BAD REGISTER ADDRESS
005142 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
005146 023737 001124 001126  CMP      $GDDAT, $BDDAT ;IS THE REGISTER ZERO
005154 001403      BEQ     64$            ;BR IF IT IS
005156 104004      EMT     4
005160 000137 006312      JMP     5$              ;BYPASS REST OF THE SUBTEST
005164 64$:
005164 113760 001224 000010  MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
005172 013737 001224 001240  MOV      PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005200 016037 000012 001126  MOV      RMD5(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
005206 042737 020001 001126  BIC     #OM!PIP, $BDDAT  ;CLEAR DONT CARE BITS
005214 012737 011600 001124  MOV      #MOL!PGM!DPR!DRY, $GDDAT ;EXPECTED STATUS
005222 013737 001124 001166  MOV      $GDDAT, $TMP1  ;USE GOOD DATA AS A MASK
005230 005137 001166      COM     $TMP1          ;COMPLEMENT THE EXPECTED STATUS
005234 013737 001126 001164  MOV      $BDDAT, $TMP0  ;SAVE THE ACTUAL STATUS
005242 043737 001166 001164  BIC     $TMP1, $TMP0    ;CLEAR UNWANTED BITS
005250 023737 001124 001164  CMP     $GDDAT, $TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
005256 001401      BEQ     65$            ;BR IF THEY ARE
005260 104005      EMT     5
005262 000240 65$: NOP

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

005264 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
005272 013737 001226 001240  MOV      PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005300 016046 000046      MOV      RMEC2(R0), -(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
005304 016046 000044      MOV      RMEC1(R0), -(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
005310 016046 000030      MOV      RMSN(R0), -(SP) ;STORE REGISTER RMSN, PORT B, FOR CHECK
005314 016046 000034      MOV      RMDC(R0), -(SP) ;STORE REGISTER RMDC, PORT B, FOR CHECK
005320 016046 000032      MOV      RMOF(R0), -(SP) ;STORE REGISTER RMOF, PORT B, FOR CHECK
005324 016046 000042      MOV      RMER2(R0), -(SP) ;STORE REGISTER RMER2, PORT B, FOR CHECK
005330 016046 000020      MOV      RMLA(R0), -(SP) ;STORE REGISTER RMLA, PORT B, FOR CHECK
005334 016046 000026      MOV      RMDT(R0), -(SP) ;STORE REGISTER RMDT, PORT B, FOR CHECK
005340 016046 000006      MOV      RMDA(R0), -(SP) ;STORE REGISTER RMDA, PORT B, FOR CHECK
    
```

```

005344 016046 000024      MOV      RMMR(R0),-(SP) ;STORE REGISTER RMMR1, PORT B, FOR CHECK
005350 016046 000014      MOV      RMER1(R0),-(SP) ;STORE REGISTER RMER1, PORT B, FOR CHECK

```

;WAIT FOR PORT A TO TIMEOUT

```

005354 005760 000012      1$:     TST      RMD5(R0) ;WAIT FOR THE DRIVE TO TIMEOUT
005360 001006                BNE      2$           ;BR WHEN TIMEOUT OCCURS
005362 005737 001260      TST      WATCH       ;CHECK WATCH
005366 001372                BNE      1$           ;BR IF NOT ZERO
005370 104036                EMT      36
005372 000137 005776      JMP      4$           ;BYPASS TIMEOUT TIME CHECK
005376 012737 000340 177776 2$:     MOV      #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
005404 013737 001256 001262      MOV      TIME,TIMEA   ;SAVE THE ELAPSED TIME FOR PORT A
005412 004537 066324      JSR      R5,TOLER     ;CALCULATE THE TOLERANCE
005416 001262                .WORD   TIMEA         ;TIMEOUT VALUE FOR PORT A
005420 012637 001264      MOV      (SP)+,TIMEAP ;+25% TOLERANCE
005424 012637 001266      MOV      (SP)+,TIMEAM ;-25% TOLERANCE

```

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

```

005430 023727 001256 000764      CMP      TIME,#500.   ;WAS MEASURED TIME AT LEAST 500 MS?
005436 103001                BHIS    3$           ;BR IF IT WAS
005440 104055                EMT      55

```

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

```

005442 012737 000240 177776 3$:     MOV      #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

005450 005037 001254                CLR      RELERR       ;CLEAR THE 'RELEASE ERROR ' INDICATOR
005454 012737 000012 001122      MOV      #RMD5,$BDADR ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
005462 060037 001122                ADD     R0,$BDADR     ;ADD THE I/O BASE ADDRESS
005466 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
005474 113760 001224 000010      MOVVB   PORTA,RMCS2(R0) ;SELECT PORT A.
005502 016037 000012 001170      MOV      RMD5(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
005510 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
005516 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
005524 042737 100100 001164      BIC     #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005532 113760 001226 000010      MOVVB   PORTB,RMCS2(R0) ;SELECT PORT B.
005540 016037 000012 001172      MOV      RMD5(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
005546 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
005554 013737 001172 001166      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
005562 042737 100100 001166      BIC     #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
005570 023737 001164 001166      CMP     $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
005576 001006                BNE     66$           ;BR IF NOT
005600 005737 001164                TST     $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
005604 001037                BNE     68$           ;BR IF NOT
005606 104046                EMT      46
005610 000137 005774                JMP     70$           ;BYPASS THE REST OF THE CHECKS
005614 013737 001170 001126 66$:     MOV      $TMP2,$BDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
005622 013737 001226 001240      MOV      PORTB,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005630 113760 001226 000010      MOVVB   PORTB,RMCS2(R0) ;SELECT PORT B.
005636 005737 001164                TST     $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
005642 001414                BEQ     67$           ;BR IF ZERO
005644 013737 001224 001240      MOV      PORTA,PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
005652 013737 001172 001126      MOV      $TMP3,$BDAT   ;'BAD DATA' FOR ERROR TYPE OUT

```

```

005660 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
005666 005737 001166                TST    $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
005672 001004                BNE    68$            ;BR IF NOT
005674 012737 177777 001254 67$:  MOV    #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
005702 104022                EMT    22
005704 013737 001170 001126 68$:  MOV    $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMD5 READ
005712 013737 001224 001240      MOV    PORTA, PTNBR   ;CHANGE PORT NUMBER
005720 042737 100100 001126      BIC    #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
005726 023737 001124 001126      CMP    $GDDAT, $BDDAT ;ALL BITS OK ?
005734 001401                BEQ    69$            ;BR IF OK FROM PORT A.
005736 104007                EMT    7
005740 013737 001172 001126 69$:  MOV    $TMP3, $BDDAT   ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
005746 013737 001226 001240      MOV    PORTB, PTNBR  ;CHANGE PORT NUMBER
005754 042737 100100 001126      BIC    #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
005762 023737 001124 001126      CMP    $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
005770 001401                BEQ    70$            ;BR IF OK
005772 104007                EMT    7
005774 000240                70$:  NOP

```

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

```

005776 013737 001226 001240 4$:  MOV    PORTB, PTNBR   ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
006004 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMER1
006010 062737 000014 001122      ADD    #RMER1, $BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
006016 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMER1
006022 001401                BEQ    .+4            ;CONTENTS ZERO ?
006024 104006                EMT    6
006026 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMMR1
006032 062737 000024 001122      ADD    #RMMR1, $BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
006040 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
006044 001401                BEQ    .+4            ;CONTENTS ZERO ?
006046 104006                EMT    6
006050 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMDA
006054 062737 000006 001122      ADD    #RMDA, $BDADR ;ADDRESS OF RMDA FOR TYPEOUT
006062 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMDA
006066 001401                BEQ    .+4            ;CONTENTS ZERO ?
006070 104006                EMT    6
006072 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMDT
006076 062737 000026 001122      ADD    #RMDT, $BDADR ;ADDRESS OF RMDT FOR TYPEOUT
006104 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMDT
006110 001401                BEQ    .+4            ;CONTENTS ZERO ?
006112 104006                EMT    6
006114 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMLA
006120 062737 000020 001122      ADD    #RMLA, $BDADR ;ADDRESS OF RMLA FOR TYPEOUT
006126 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMLA
006132 001401                BEQ    .+4            ;CONTENTS ZERO ?
006134 104006                EMT    6
006136 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMER2
006142 062737 000042 001122      ADD    #RMER2, $BDADR ;ADDRESS OF RMER2 FOR TYPEOUT
006150 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMER2
006154 001401                BEQ    .+4            ;CONTENTS ZERO ?
006156 104006                EMT    6
006160 010037 001122                MOV    R0, $BDADR    ;BASE ADDRESS FOR REGISTER RMOF
006164 062737 000032 001122      ADD    #RMOF, $BDADR ;ADDRESS OF RMOF FOR TYPEOUT
006172 012637 001126                MOV    (SP)+, $BDDAT ;CHECK THE STORED CONTENTS OF RMOF
006176 001401                BEQ    .+4            ;CONTENTS ZERO ?

```

```

006200 104006 EMT 6
006202 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDC
006206 062737 000034 001122 ADD #RMDC,$BDADR ;ADDRESS OF RMDC FOR TIMEOUT
006214 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDC
006220 001401 BEQ .+4 ;CONTENTS ZERO ?
006222 104006 EMT 6
006224 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMSN
006230 062737 000030 001122 ADD #RMSN,$BDADR ;ADDRESS OF RMSN FOR TIMEOUT
006236 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMSN
006242 001401 BEQ .+4 ;CONTENTS ZERO ?
006244 104006 EMT 6
006246 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC1
006252 062737 000044 001122 ADD #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TIMEOUT
006260 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC1
006264 001401 BEQ .+4 ;CONTENTS ZERO ?
006266 104006 EMT 6
006270 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC2
006274 062737 000046 001122 ADD #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TIMEOUT
006302 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC2
006306 001401 BEQ .+4 ;CONTENTS ZERO ?
006310 104006 EMT 6
006312 000004 5$: SCOPE ;LOOP ?

```

152
170
171

```

*****
*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.
*****

```

```

006314 005737 001300 TST3: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
006320 001406 BEQ 2$ ;BR IF NOT
006322 100002 BPL 1$ ;BR IF JUST ENTERED TEST
006324 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
006330 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
006336 012737 006352 001106 2$: MOV #TEST3,$LPADR ;SETUP SCOPE LOOP ADDRESS
006344 012737 006352 001110 MOV #TEST3,$LPERR ;SETUP ERROR LOOP ADDRESS
006352 TEST3:
006352 112737 000003 001102 MOVB #3,$TSTNM ;MOVE #3 TO TEST NUMBER
006360 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
006364 012737 000002 001176 MOV #2,$TIMES ;DO 2. ITERATIONS
172 006372 012737 000240 177776 MOV #<5*32.>,@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
173 006400 005037 001270 CLR TIMEB ;CLEAR TIMEOUT VALUE FOR PORT B

```

006404 005037 001272 CLR TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
 006410 005037 001274 CLR TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

006414 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
 006420 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT B

006426 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 006434 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
 006442 005060 000006 CLR RMDA(R0) ;WRITE RMDA
 006446 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
 006454 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006462 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
 006470 016037 000012 001126 MOV RMD5(R0), \$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
 006476 010037 001122 MOV R0, \$BDADR ;RH/RM BASE ADDRESS
 006502 062737 000012 001122 ADD #RMD5, \$BDADR ;GENERATE BAD REGISTER ADDRESS
 006510 005037 001124 CLR \$GDDAT ;REGISTER SHOULD BE ZERO
 006514 023737 001124 001126 CMP \$GDDAT, \$BDDAT ;IS THE REGISTER ZERO
 006522 001403 BEQ 64\$;BR IF IT IS
 006524 104004 EMT 4
 006526 000137 007660 JMP 5\$;BYPASS REST OF THE SUBTEST

64\$:

006532 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 006540 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006546 016037 000012 001126 MOV RMD5(R0), \$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
 006554 042737 020001 001126 BIC #OM!PIP, \$BDDAT ;CLEAR DONT CARE BITS
 006562 012737 011600 001124 MOV #MOL!PGM!DPR!DRY, \$GDDAT ;EXPECTED STATUS
 006570 013737 001124 001166 MOV \$GDDAT, \$TMP1 ;USE GOOD DATA AS A MASK
 006576 005137 001166 COM \$TMP1 ;COMPLEMENT THE EXPECTED STATUS
 006602 013737 001126 001164 MOV \$BDDAT, \$TMP0 ;SAVE THE ACTUAL STATUS
 006610 043737 001166 001164 BIC \$TMP1, \$TMP0 ;CLEAR UNWANTED BITS
 006616 023737 001124 001164 CMP \$GDDAT, \$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
 006624 001401 BEQ 65\$;BR IF THEY ARE
 006626 104005 EMT 5
 006630 000240 NOP

65\$:

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

006632 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
 006640 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006646 016046 000046 MOV RMEC2(R0), -(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
 006652 016046 000044 MOV RMEC1(R0), -(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
 006656 016046 000030 MOV RMSN(R0), -(SP) ;STORE REGISTER RMSN, PORT A, FOR CHECK
 006662 016046 000034 MOV RMDC(R0), -(SP) ;STORE REGISTER RMDC, PORT A, FOR CHECK
 006666 016046 000032 MOV RMOF(R0), -(SP) ;STORE REGISTER RMOF, PORT A, FOR CHECK
 006672 016046 000042 MOV RMER2(R0), -(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
 006676 016046 000020 MOV RMLA(R0), -(SP) ;STORE REGISTER RMLA, PORT A, FOR CHECK
 006702 016046 000026 MOV RMDT(R0), -(SP) ;STORE REGISTER RMDT, PORT A, FOR CHECK
 006706 016046 000006 MOV RMDA(R0), -(SP) ;STORE REGISTER RMDA, PORT A, FOR CHECK
 006712 016046 000024 MOV RMMR1(R0), -(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
 006716 016046 000014 MOV RMER1(R0), -(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK

;WAIT FOR PORT B TO TIMEOUT


```

006722 005760 000012 1$: TST RMDS(R0) ;WAIT FOR THE DRIVE TO TIMEOUT
006726 001006 BNE 2$ ;BR WHEN TIMEOUT OCCURS
006730 005737 001260 TST WATCH ;CHECK WATCH
006734 001372 BNE 1$ ;BR IF NOT ZERO
006736 104036 EMT 36
006740 000137 007344 JMP 4$ ;BYPASS TIMEOUT TIME CHECK
006744 012737 000340 177776 2$: MOV #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
006752 013737 001256 001270 MOV TIME,TIMEB ;SAVE THE ELAPSED TIME FOR PORT B
006760 004537 066324 JSR R5,TOLER ;CALCULATE THE TOLFRANCE
006764 001270 .WORD TIMEB ;TIMEOUT VALUE FOR PORT B
006766 012637 001272 MOV (SP)+,TIMEBP ;+25% TOLERANCE
006772 012637 001274 MOV (SP)+,TIMEBM ;-25% TOLERANCE

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

006776 023727 001256 000764 CMP TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
007004 103001 BHIS 3$ ;BR IF IT WAS
007006 104055 EMT 55

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

007010 012737 000240 177776 3$: MOV #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

007016 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
007022 012737 000012 001122 MOV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
007030 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
007034 012737 011600 1124 MOV #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
007042 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
007050 016037 000012 001170 MOV RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
007056 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
007064 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
007072 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007100 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
007106 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
007114 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
007122 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
007130 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007136 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
007144 001006 BNE 66$ ;BR IF NOT
007146 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
007152 001037 BNE 68$ ;BR IF NOT
007154 104046 EMT 46
007156 000137 007342 JMP 70$ ;BYPASS THE REST OF THE CHECKS
007162 013737 001170 001126 66$: MOV $TMP2,$BDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
007170 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007176 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
007204 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
007210 001414 BEQ 67$ ;BR IF ZERO
007212 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007220 013737 001172 001126 MOV $TMP3,$BDAT ;'BAD DATA' FOR ERROR TYPE OUT
007226 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
007234 005737 001164 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
007240 001004 BNE 68$ ;BR IF NOT
007242 012737 177777 001254 67$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
007250 104022 EMT 22
  
```

```

007252 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
007260 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
007266 042737 100100 001126 BIC #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007274 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
007302 001401 BEQ 69$ ;BR IF OK FROM PORT A.
007304 104007 EMT 7
007306 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
007314 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
007322 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007330 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
007336 001401 BEQ 70$ ;BR IF OK
007340 104007 EMT 7
007342 000240 70$: NOP
  
```

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

```

007344 013737 001224 001240 4$: MOV PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007352 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER1
007356 062737 000014 001122 ADD #RMER1,$BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
007364 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER1
007370 001401 BEQ .+4 ;CONTENTS ZERO ?
007372 104006 EMT 6
007374 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMMR1
007400 062737 000024 001122 ADD #RMMR1,$BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
007406 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
007412 001401 BEQ .+4 ;CONTENTS ZERO ?
007414 104006 EMT 6
007416 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
007422 062737 000006 001122 ADD #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
007430 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
007434 001401 BEQ .+4 ;CONTENTS ZERO ?
007436 104006 EMT 6
007440 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
007444 062737 000026 001122 ADD #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
007452 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
007456 001401 BEQ .+4 ;CONTENTS ZERO ?
007460 104006 EMT 6
007462 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMLA
007466 062737 000020 001122 ADD #RMLA,$BDADR ;ADDRESS OF RMLA FOR TYPEOUT
007474 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMLA
007500 001401 BEQ .+4 ;CONTENTS ZERO ?
007502 104006 EMT 6
007504 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER2
007510 062737 000042 001122 ADD #RMER2,$BDADR ;ADDRESS OF RMER2 FOR TYPEOUT
007516 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER2
007522 001401 BEQ .+4 ;CONTENTS ZERO ?
007524 104006 EMT 6
007526 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMOF
007532 062737 000032 001122 ADD #RMOF,$BDADR ;ADDRESS OF RMOF FOR TYPEOUT
007540 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMOF
007544 001401 BEQ .+4 ;CONTENTS ZERO ?
007546 104006 EMT 6
007550 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDC
007554 062737 000034 001122 ADD #RMDC,$BDADR ;ADDRESS OF RMDC FOR TYPEOUT
007562 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDC
007566 001401 BEQ .+4 ;CONTENTS ZERO ?
  
```

```

007570 104006 EMT 6
007572 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMSN
007576 062737 000030 001122 ADD #RMSN,$BDADR ;ADDRESS OF RMSN FOR TYPEOUT
007604 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMSN
007610 001401 BEQ .+4 ;CONTENTS ZERO ?
007612 104006 EMT 6
007614 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC1
007620 062737 000044 001122 ADD #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TYPEOUT
007626 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC1
007632 001401 BEQ .+4 ;CONTENTS ZERO ?
007634 104006 EMT 6
007636 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC2
007642 062737 000046 001122 ADD #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TYPEOUT
007650 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC2
007654 001401 BEQ .+4 ;CONTENTS ZERO ?
007656 104006 EMT 6
007660 000004 5$: SCOPE ;LOOP ?

```

174
188
189

```

:*****
:*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.
:*****

```

```

007662 005737 001300 TST4: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
007666 001406 BEQ 2$ ;BR IF NOT
007670 100002 BPL 1$ ;BR IF JUST ENTERED TEST
007672 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
007676 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
007704 012737 007720 001106 2$: MOV #TEST4,$LPADR ;SETUP SCOPE LOOP ADDRESS
007712 012737 007720 001110 MOV #TEST4,$LPERR ;SETUP ERROR LOOP ADDRESS
007720 112737 000004 001102 TEST4: MOVB #4,$STNUM ;MOVE #4 TO TEST NUMBER
007726 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
007732 012737 000012 001176 MOV #10,$TIMES ;;DO 10. ITERATIONS

```

190
223

;START THE TIMER

```

007740 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
007744 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000. MS

```

;SEIZE THE DRIVE AND SET VOLUME VALID

;SEIZE THE DRIVE THROUGH PORT A

```

007752 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A

```

```

007760 013737 001224 001242      MOV    PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
007766 005060 000012                CLR    RMDS(R0)      ;WRITE RMDS
007772 013737 001226 001244      MOV    PORTB,OPPRT  ;'OPPOSITE' PORT ADDRESS
010000 012760 000021 000000      MOV    #21,RMCS1(R0) ;SET VOLUME VALID
010006 005037 001250                CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
010012 016037 000012 001126      MOV    RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
010020 012737 000012 001122      MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010026 060037 001122                ADD    R0,$BDADR    ;ADD RH/RM BASE ADDRESS
010032 012737 000100 001124      MOV    #VV,$GDDAT   ;WHAT REGISTER SHOULD BE
010040 013737 001126 001164      MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010046 042737 177677 001164      BIC    #^CVV,$TMP0  ;SAVE SPECIFIED BITS
010054 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
010062 001414                BEQ    66$          ;BR IF OK
010064 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
010072 042737 000100 001174      BIC    #VV,$TMP4    ;CLEAR THE MASKED BITS
010100 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010106 104013                EMT    13
010110 005137 001250                COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
010114 000240                NOP
010116 012760 000040 000010      MOV    #CLR,RMCS2(R0) ;CLEAR DRIVE

;RELEASE THE DRIVE FROM PORT A

010124 113760 001224 000010      MOV    PORTA,RMCS2(R0) ;SELECT PORT A
010132 013737 001224 001240      MOV    PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010140 012760 000013 000000      MOV    #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

010146 005037 001254                CLR    RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
010152 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
010160 060037 001122                ADD    R0,$BDADR    ;ADD THE I/O BASE ADDRESS
010164 012737 011600 001124      MOV    #MGL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
010172 113760 001224 000010      MOV    PORTA,RMCS2(R0) ;SELECT PORT A.
010200 016037 000012 001170      MOV    RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
010206 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
010214 013737 001170 001164      MOV    $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
010222 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010230 113760 001226 000010      MOV    PORTB,RMCS2(R0) ;SELECT PORT B.
010236 016037 000012 001172      MOV    RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
010244 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
010252 013737 001172 001166      MOV    $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
010260 042737 100100 001166      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010266 023737 001164 001166      CMP    $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
010274 001006                BNE    68$         ;BR IF NOT
010276 005737 001164                TST    $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
010302 001037                BNE    70$         ;BR IF NOT
010304 104046                EMT    46
010306 000137 010472                JMP    72$         ;BYPASS THE REST OF THE CHECKS
010312 013737 001170 001126      MOV    $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
010320 013737 001226 001240      MOV    PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010326 113760 001226 000010      MOV    PORTB,RMCS2(R0) ;SELECT PORT B.
010334 005737 001164                TST    $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
010340 001414                BEQ    69$         ;BR IF ZERO
010342 013737 001224 001240      MOV    PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010350 013737 001172 001126      MOV    $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
010356 113760 001224 000010      MOV    PORTA,RMCS2(R0) ;SELECT PORT A.
  
```

```

010364 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
010370 001004      BNE      70$      ;BR IF NOT
010372 012737 177777 001254 69$:  MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
010400 104022      EMT      22
010402 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
010410 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
010416 042737 100100 001126      BIC      #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
010424 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
010432 001401      BEQ      71$      ;BR IF OK FROM PORT A.
010434 104007      EMT      7
010436 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
010444 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
010452 042737 100100 001126      BIC      #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
010460 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
010466 001401      BEQ      72$      ;BR IF OK
010470 104007      EMT      7
010472 000240      NOP
010474 005737 001254      TST      RELEHR      ;DID DRIVE RETURN TO NEUTRAL ?
010500 001402      BEQ      +6        ;BR IF IN NEUTRAL
010502 000137 010756      JMP      1$        ;GO WAIT FOR DRIVE TO TIMEOUT
010506 113760 001224 000010      MOV      PORTA, RMCS2(RO) ;SELECT PORT A
010514 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010522 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
010526 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
010534 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010542 060037 001122      ADD      RO,$BDADR   ;ADD RH/RM BASE ADDRESS
010546 005037 001124      CLR      $GDDAT     ;WHAT REGISTER SHOULD BE
010552 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010560 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
010566 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
010574 001414      BEQ      73$      ;BR IF OK
010576 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
010604 042737 100000 001174      BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
010612 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010620 104017      EMT      17
010622 005137 001250      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
010626 000240      NOP
010630 113760 001226 000010      MOV      PORTB, RMCS2(RO) ;SELECT PORT B
010636 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010644 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
010650 016037 000012 001126      MOV      RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
010656 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010664 060037 001122      ADD      RO,$BDADR   ;ADD RH/RM BASE ADDRESS
010670 005037 001124      CLR      $GDDAT     ;WHAT REGISTER SHOULD BE
010674 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010702 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
010710 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
010716 001414      BEQ      75$      ;BR IF OK
010720 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
010726 042737 100000 001174      BIC      #ATA,$TMP4   ;CLEAR THE MASKED BITS
010734 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010742 104017      EMT      17
010744 005137 001250      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
010750 000240      NOP
010752 000137 011010      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

```

010756      113760 001226 000010      1$:      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
010756      013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010764      005760 000012                TST   RMDS(R0) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
010772      001004                BNE   2$ ;BR WHEN DRIVE RELEASED
010776      005737 001260                TST   WATCH ;CHECK THE WATCH
011000      001364                BNE   1$ ;BR IF NOT ZERO
011004      104036                EMT   36
011006      000004                2$:   SCOPE ;LOOP ?
011010
    
```

224
238
239

```

*****
*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.
*****
    
```

```

011012      005737 001300      TST5:   TST   KYBCTL ;PERFORMING ONLY SINGLE TEST ?
011012      001406                BEQ   2$ ;BR IF NOT
011016      100002                BPL   1$ ;BR IF JUST ENTERED TEST
011020      000137 003074                JMP   EXEC ;RETURN & GET NEXT TEST NUMBER
011022      012737 177777 001300      1$:   MOV   #-1, KYBCTL ;SET SINGLE TEST INDICATOR
011026      012737 011050 001106      2$:   MOV   #TEST5, $LPADR ;SETUP SCOPE LOOP ADDRESS
011034      012737 011050 001110      MOV   #TEST5, $LPERR ;SETUP ERROR LOOP ADDRESS
011042      112737 000005 001102      TEST5: MOVB  #5, $STSTM ;MOVE #5 TO TEST NUMBER
011050      012706 001100                MOV   #STACK, SP ;LOAD THE STACK POINTER
011056      012737 000012 001176      MOV   #10., $TIMES ;DO 10. ITERATIONS
011062
    
```

240
241

;START THE TIMER

```

011070      005037 001256      CLR   TIME ;CLEAR THE ELAPSED TIME COUNTER
011074      012737 003720 001260      MOV   #2000., WATCH ;SET WATCH TO 2000. MS
    
```

;SEIZE THE DRIVE AND SET VOLUME VALID

;SEIZE THE DRIVE THROUGH PORT B

```

011102      113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
011110      013737 001226 001242      MOV   PORTB, SEIZPT ;STORF SEIZING PORT'S ADDRESS
011116      005060 000012                CLR   RMDS(R0) ;WRITE RMDS
011122      013737 001224 001244      MOV   PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
011130      012760 000021 000000      MOV   #21, RMCS1(R0) ;SET VOLUME VALID
011136      005037 001250                CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
011142      016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
    
```

```

011150 012737 000012 001122      MOV      #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011156 060037 001122      ADD      R0,$BDADR   ;ADD RH/RM BASE ADDRESS
011162 012737 000100 001124      MOV      #VV,$GDDAT ;WHAT REGISTER SHOULD BE
011170 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011176 042737 177677 001164      BIC      #^LVV,$TMP0 ;SAVE SPECIFIED BITS
011204 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
011212 001414      BEQ      66$        ;BR IF OK
011214 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
011222 042737 000100 001174      BIC      #VV,$TMP4   ;CLEAR THE MASKED BITS
011230 053737 001174 001124      BIS      $TMP4,$GDDA1 ;'OR' WITH GOOD DATA FOR TYPEOUT
011236 104013      EMT      13
011240 005137 001250      COM      CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
011244 000240      NOP
011246 012760 000040 000010 66$: MOV      #CLR,RMCS2(R0) ;CLEAR DRIVE
  
```

;RELEASE THE DRIVE FROM PORT B

```

011254 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
011262 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011270 012760 000013 000000      MOV      #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

011276 005037 001254      CLR      RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
011302 012737 000012 001122      MOV      #RMD5,$BDADR ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
011310 060037 001122      ADD      R0,$BDADR   ;ADD THE I/O BASE ADDRESS
011314 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
011322 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
011330 016037 000012 001170      MOV      RMD5(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
011336 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
011344 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
011352 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011360 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
011366 016037 000012 001172      MOV      RMD5(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
011374 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
011402 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
011410 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011416 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
011424 001006      BNE      68$        ;BR IF NOT
011426 005737 001164      TST      $TMP0       ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
011432 001037      BNE      70$        ;BR IF NOT
011434 104046      EMT      46
011436 000137 011622      JMP      72$
011442 013737 001170 001126 68$: MOV      $TMP2,$BDDAT ;BYPASS THE REST OF THE CHECKS
011450 013737 001226 001240      MOV      PORTB,PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
011456 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011464 005737 001164      TST      $TMP0       ;SELECT PORT B.
011470 001414      BEQ      69$        ;SEE IF STATUS EQ 0 FROM PORT A.
011472 013737 001224 001240      MOV      PORTA,PTNBR ;BR IF ZERO
011500 013737 001172 001126      MOV      $TMP3,$BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011506 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;'BAD DATA' FOR ERROR TYPE OUT
011514 005737 001166      TST      $TMP1       ;SELECT PORT A.
011520 001004      BNE      70$        ;SEE IF STATUS EQ ZERO FROM PORT B.
011522 012737 177777 001254 69$: MOV      #-1,RELERR ;BR IF NOT
011530 104022      EMT      22 ;SET 'RELEASE ERROR' INDICATOR
011532 013737 001170 001126 70$: MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
011540 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
  
```



```

011546 042737 100100 001126 BIC #ATA,VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011554 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
011562 001401 BEQ 71$ ;BR IF OK FROM PORT A.
011564 104007 EMT 7
011566 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
011574 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
011602 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011610 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
011616 001401 BEQ 72$ ;BR IF OK
011620 104007 EMT 7
011622 000240 72$: NOP
011624 005737 001254 TST RELERR ;DID DRIVE RETURN TO NEUTRAL ?
011630 001402 BEQ +6 ;BR IF IN NEUTRAL
011632 000137 012106 JMP 1$ ;GO WAIT FOR DRIVE TO TIMEOUT
011636 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
011644 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011652 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
011656 016037 000012 001126 MOV RMDs(R0),$BDDAT ;GET CONTENTS OF RMDs
011664 012737 000012 001122 MOV #RMDs,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011672 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
011676 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
011702 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011710 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
011716 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
011724 001414 BEQ 73$ ;BR IF OK
011726 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
011734 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
011742 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
011750 104017 EMT 17
011752 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
011756 000240 73$: NOP
011760 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
011766 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011774 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
012000 016037 000012 001126 MOV RMDs(R0),$BDDAT ;GET CONTENTS OF RMDs
012006 012737 000012 001122 MOV #RMDs,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
012014 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
012020 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
012024 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
012032 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
012040 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
012046 001414 BEQ 75$ ;BR IF OK
012050 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
012056 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
012064 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
012072 104017 EMT 17
012074 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
012100 000240 75$: NOP
012102 000137 012140 JMP 2$ ;GO CHECK FOR LOOP ON ERROR

```

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

```

012106 1$: MOVB PORTA,RMCS2(R0) ;SELECT PORT A
012106 113760 001224 000010 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012114 013737 001224 001240 TST RMDs(R0) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
012122 005760 000012

```

012126 001004
 012130 005737 001260
 012134 001364
 012136 104036
 012140 000004

BNE 2\$;BR WHEN DRIVE RELEASED
 TST WATCH ;CHECK THE WATCH
 BNE 1\$;BR IF NOT ZERO
 EM 36
 2\$: SCOPE ;LOOP ?

246
 255
 256

 *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.

012142
 012142 005737 001300
 012146 001406
 012150 100002
 012152 000137 003074
 012156 012737 177777 001300
 012164 012737 012200 001106
 012172 012737 012200 001110
 012200
 012200 112737 000006 001102
 012206 012706 001100
 012212 012737 000012 001176

TST6:
 TST KYBCTL ;PERFORMING ONLY SINGLF 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 #TEST6,\$LPADR ;SETUP SCOPE LOOP ADDRESS
 MOV #TEST6,\$LPERR ;SETUP ERROR LOOP ADDRESS
 TEST6:
 MOVB #6,\$STSTM ;MOVE #6 TO TEST NUMBER
 MOV #STACK,SP ;LOAD THE STACK POINTER
 MOV #10,\$TIMES ;DO 10. ITERATIONS
 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 MOV PORTA, SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

257
 268

012220 113760 001224 000010
 012226 013737 001224 001240
 012234 013737 001224 001242

;ISSUE A RELEASE COMMAND
 MOV #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

012250 005037 001254
 012254 012737 000012 001122
 012262 060037 001122
 012266 012737 011700 001124
 012274 113760 001224 000010
 012302 016037 000012 001170
 012310 042737 024001 001170
 012316 013737 001170 001164
 012324 042737 100100 001164
 012332 113760 001226 000010
 012340 016037 000012 001172
 012346 042737 024001 001172
 012354 013737 001172 001166
 012362 042737 100100 001166
 012370 023737 001164 001166
 012376 001006
 012400 005737 001164
 012404 001045
 012406 104046

CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 MOV #RMD5,\$BDADR ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
 ADD R0,\$BDADR ;ADD THE I/O BASE ADDRESS
 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
 MOVB PORTA, RMCS2(R0) ;SELECT PORT A.
 MOV RMD5(R0), \$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 BIC #PIP!WRL!OM,\$TMP2 ;CLEAR DONT CARES
 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 MOVB PORTB, RMCS2(R0) ;SELECT PORT B.
 MOV RMD5(R0), \$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 BIC #PIP!WRL!OM,\$TMP3 ;CLEAR DONT CARES
 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 BNE 64\$;BR IF NOT
 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 BNE 66\$;BR IF NOT
 EMT 46

```

012410 000137 012610          JMP      68$          ;BYPASS THE REST OF THE CHECKS
012414 013737 001170 001126 64$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012422 013737 001226 001240      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012430 113760 001226 000010      MOVVB  PORTB,RMCS2(R0) ;SELECT PORT B.
012436 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
012442 001414          BEQ     65$          ;BR IF ZERO
012444 013737 001224 001240      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012452 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
012460 113760 001224 000010      MOVVB  PORTA,RMCS2(R0) ;SELECT PORT A.
012466 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
012472 001012          BNE     66$          ;BR IF NOT
012474 012737 177777 001254 65$:  MOV     #-1,RELEERR  ;SET 'RELEASE ERROR' INDICATOR
012502 012760 000011 000000      MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
012510 012760 000013 000000      MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
012516 104030          EMT     30
012520 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD'S READ
012526 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
012534 042737 100000 001126      BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
012542 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
012550 001401          BEQ     67$          ;BR IF OK FROM PORT A.
012552 104007          EMT     7
012554 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT ;CHECK RMD'S FOR BIT FAILURES - FROM PORT B.
012562 013737 001226 001240      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
012570 042737 100000 001126      BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
012576 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
012604 001401          BEQ     68$          ;BR IF OK
012606 104007          EMT     7
012610 000240          NOP
012612 000004          SCOPE          ;LOOP ?
    
```

269
278
279

```

*****
*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.
*
*****
    
```

```

012614          TST7:
012614 005737 001300          TST     KYBCTL        ;PERFORMING ONLY SINGLE TEST ?
012620 001406          BEQ     2$           ;BR IF NOT
012622 100002          BPL     1$           ;BR IF JUST ENTERED TEST
012624 000137 003074          JMP     EXEC         ;RETURN & GET NEXT TEST NUMBER
012630 012737 177777 001300 1$:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
012636 012737 012652 001106 2$:  MOV     #TEST7,$LPADR ;SETUP SCOPE LOOP ADDRESS
012644 012737 012652 001110      MOV     #TEST7,$LPERR ;SETUP ERROR LOOP ADDRESS
012652          TEST7:
012652 112737 000007 001102      MOVVB  #7,$STNM      ;MOVE #7 TO TEST NUMBER
012660 012706 001100          MOV     #STACK,SP   ;LOAD THE STACK POINTER
012664 012737 000012 001176      MOV     #10.,$TIMES ;DO 10. ITERATIONS
280
281 012672 113760 001226 000010      MOVVB  PORTB,RMCS2(R0) ;SELECT PORT B
012700 013737 001226 001240      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012706 013737 001226 001242      MOV     PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
    
```

```

012714 012760 000013 000000 ;ISSUE A RELEASE COMMAND
                                MOV #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND
                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

012722 005037 001254          CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
012726 012737 000012 001122 MOV #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
012734 060037 001122          ADD R0, $BDADR ;ADD THE I/O BASE ADDRESS
012740 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
012746 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A.
012754 016037 000012 001170 MOV RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012762 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
012770 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
012776 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013004 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B.
013012 016037 000012 001172 MOV RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
013020 042737 024001 001172 BIC #PIP.WRL!OM, $TMP3 ;CLEAR DONT CARES
013026 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
013034 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013042 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
013050 001006          BNE 64$ ;BR IF NOT
013052 005737 001164          TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
013056 001045          BNE 66$ ;BR IF NOT
013060 104046          EMT 46
013062 000137 013262          JMP 68$ ;BYPASS THE REST OF THE CHECKS
013066 013737 001170 001126 64$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
013074 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013102 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B.
013110 005737 001164          TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
013114 001414          BEQ 65$ ;BR IF ZERO
013116 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013124 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
013132 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A.
013140 005737 001166          TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
013144 001012          BNE 66$ ;BR IF NOT
013146 012737 177777 001254 65$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
013154 012760 000011 000000 MOV #11, RMCS1(R0) ;CLEAR THE DRIVE
013162 012760 000013 000000 MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
013170 104030          EMT 30
013172 013737 001170 001126 66$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
013200 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
013206 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
013214 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
013222 001401          BEQ 67$ ;BR IF OK FROM PORT A.
013224 104007          EMT 7
013226 013737 001172 001126 67$: MOV $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
013234 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
013242 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
013250 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
013256 001401          BEQ 68$ ;BR IF OK
013260 104007          EMT 7
013262 000240          NOP
013264 000004          SCOPE ;LOOP ?
  
```

282
301
302

 ;*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.

013266
 013266 005737 001300
 013272 001406
 013274 100002
 013276 000137 003074
 013302 012737 177777 001300
 013310 012737 013324 001106
 013316 012737 013324 001110
 013324
 013324 112737 000010 001102
 013332 012706 001100
 013336 012737 000012 001176

```
TST10:
  ^ST      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      #TEST10,$LPADR ;SETUP SCOPE LOOP ADDRESS
         MOV      #TEST10,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST10:
         MOVB     #10,$STNM      ;MOVE #10 TO TEST NUMBER
         MOV      #STACK,SP     ;LOAD THE STACK POINTER
         MOV      #10,,$TIMES   ;;DO 10. ITERATIONS
```

303
 327

;CLEAR ATTENTION BITS FOR BOTH PORTS

013344 113760 001224 000010
 013352 005060 000012
 013356 012760 000011 000000
 013364 012760 000013 000000
 013372 113760 001226 000010
 013400 005060 000012
 013404 012760 000011 000000
 013412 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

013420 113760 001226 000010
 013426 013737 001226 001242
 013434 005060 000012
 013440 113760 001224 000010
 013446 013737 001224 001240
 013454 013737 001224 001244
 013462 016037 000012 001126
 013470 010037 001122
 013474 062737 000012 001122
 013502 005037 001124
 013506 023737 001124 001126
 013514 001403
 013516 104004

```
         MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
         MOV      PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
         CLR      RMDS(R0)        ;WRITE RMDS
         MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
         MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
         MOV      PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS
         MOV      RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
         MOV      R0,$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
```

```

013520 000137 014530          JMP      1$          ;BYPASS REST OF THE SUBTEST
013524          64$:   MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
013524 113760 001226 000010    MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013532 013737 001226 001240    MOV   RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
013540 016037 000012 001126    BIC   #OM.PIP, $BDDAT ;CLEAR DONT CARE BITS
013546 042737 020001 001126    MOV   #MOL.PGM.DPR.DRY!VV, $GDDAT ;EXPECTED STATUS
013554 012737 011700 001124    MOV   $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
013562 013737 001124 001166    COM   $TMP1 ;COMPLEMENT THE EXPECTED STATUS
013570 005137 001166          MOV   $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
013574 013737 001126 001164    BIC   $TMP1, $TMP0 ;CLEAR UNWANTED BITS
013602 043737 001166 001164    CMP   $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
013610 023737 001124 001164    BEQ   65$          ;BR IF THEY ARE
013616 001401          EMT   5
013620 104005          65$:   NOP
013622 000240
  
```

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

```

013624 113760 001224 000010    MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
013632 013737 001224 001240    MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013640 012760 000013 000000    MOV   #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

```

013646 005037 001250          CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
013652 016037 000012 001126    MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
013660 012737 000012 001122    MOV   #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
013666 060037 001122          ADD   R0, $BDADR ;ADD RH/RM BASE ADDRESS
013672 005037 001124          CLR   $GDDAT ;WHAT REGISTER SHOULD BE
013676 023737 001124 001126    CMP   $GDDAT, $BDDAT ;IS THE REGISTER OK ?
013704 001403          BEQ   66$          ;BR IF OK
013706 104010          EMT   10
013710 005137 001250          COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
013714 016037 000000 001126    66$:   MOV   RMCS1(R0), $BDDAT ;GET THE CONTENTS OF RMCS1
013722 012737 000000 001122    MOV   #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
013730 060037 001122          ADD   R0, $BDADR ;ADDRESS BASE
013734 032737 020000 001126    BIT   #MCPE, $BDDAT ;IS 'MCPE' SET ?
013742 001404          BEQ   67$          ;BR IF NOT
013744 104011          EMT   11
013746 012760 040000 000000    67$:   MOV   #TRE, RMCS1(R0) ;CLEAR 'MCPE'
013754 000240          NOP
013756 005737 001250          TST   CKERR ;WAS RMDS NON ZERO ?
013762 001402          BEQ   +6 ;CONTENTS OF RMDS SEEN BY PORT A
013764 000137 014530          JMP   1$ ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
  
```

;RELEASE THE DRIVE FROM PORT B

```

013770 113760 001226 000010    MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
013776 013737 001226 001240    MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014004 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

```

014012 005037 001254          CLR   RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
014016 012737 111700 001124    MOV   #ATA!MOL.PGM.DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
014024 012737 000012 001122    MOV   #RMDS, $BDADR ;REGISTER ADDRESS INCREMENT
014032 060037 001122          ADD   R0, $BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
  
```

```

014036 113760 001224 000010      MOVB   PORTA, RMCS2(R0)      ; SELECT PORT A
014044 013737 001224 001240      MOV    PORTA, PTNBR        ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014052 016037 000012 001164      MOV    RMDS(R0), $TMP0     ; READ STATUS REGISTER FROM PORT A
014060 113760 001226 000010      MOVB   PORTB, RMCS2(R0)    ; SELECT PORT B
014066 013737 001226 001240      MOV    PORTB, PTNBR        ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014074 016037 000012 001126      MOV    RMDS(R0), $BDDAT    ; DRIVE STATUS FROM PORT B
014102 001404                      BEQ    68$                 ; BR IF STATUS FROM PORT B ZERO
014104 005737 001164                      TST    $TMP0               ; IS STATUS FROM PORT A ZERO ?
014110 001401                      BEQ    68$                 ; BR IF ZERO
014112 104031                      EMT    31
014114 013737 001164 001126 68$:  MOV    $TMP0, $BDDAT       ; CHECK STATUS FROM PORT A
014122 013737 001224 001240      MOV    PORTA, PTNBR        ; CHANGE PORT ADDRESS FOR TYPEOUT
014130 023737 001124 001126      CMP    $GDDAT, $BDDAT     ; COMPARE WITH CONSTANT
014136 001401                      BEQ    69$                 ; BR IF OK
014140 104027                      EMT    27
014142 000240                      69$:  NOP

;RELEASE THE DRIVE FROM PORT A

C14144 113760 001224 000010      MOVB   PORTA, RMCS2(R0)    ; SELECT PORT A
014152 013737 001224 001240      MOV    PORTA, PTNBR        ; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014160 012760 000013 000000      MOV    #13, RMCS1(R0)     ; ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

014166 005037 001254                      CLR    RELERR              ; CLEAR THE 'RELEASE ERROR' INDICATOR
014172 012737 000012 001122      MOV    #RMDS, $BDADR      ; FORM THE ADDRESS OF RMDS FOR TYPEOUT
014200 060037 001122                      ADD    R0, $BDADR         ; ADD THE I/O BASE ADDRESS
014204 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ; COMPARISON CONSTANT
014212 113760 001224 000010      MOVB   PORTA, RMCS2(R0)    ; SELECT PORT A.
014220 016037 000012 001170      MOV    RMDS(R0), $TMP2    ; GET THE DRIVE STATUS REGISTER FROM PORT A.
014226 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ; CLEAR DONT CARES
014234 013737 001170 001164      MOV    $TMP2, $TMP0       ; COPY IT INTO '$TMP0'
014242 042737 100100 001164      BIC    #ATA!VV, $TMP0     ; CLEAR PORT DEPENDENT BITS FROM THE COPY
014250 113760 001226 000010      MOVB   PORTB, RMCS2(R0)    ; SELECT PORT B.
014256 016037 000012 001172      MOV    RMDS(R0), $TMP3    ; GET THE DRIVE STATUS REGISTER FROM PORT B.
014264 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ; CLEAR DONT CARES
014272 013737 001172 001166      MOV    $TMP3, $TMP1       ; COPY IT INTO '$TMP1'
014300 042737 100100 001166      BIC    #ATA!VV, $TMP1     ; CLEAR PORT DEPENDENT BITS FROM THE COPY
014306 023737 001164 001166      CMP    $TMP0, $TMP1       ; IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
014314 001006                      BNE    70$                 ; BR IF NOT
014316 005737 001164                      TST    $TMP0               ; REGISTERS ARE THE SAME: ARE THEY ZERO ?
014322 001045                      BNE    72$                 ; BR IF NOT
014324 104046                      EMT    46
014326 000137 014526                      JMP    74$                 ; BYPASS THE REST OF THE CHECKS
014332 013737 001170 001126 70$:  MOV    $TMP2, $BDDAT       ; SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014340 013737 001226 001240      MOV    PORTB, PTNBR        ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014346 113760 001226 000010      MOVB   PORTB, RMCS2(R0)    ; SELECT PORT B.
014354 005737 001164                      TST    $TMP0               ; SEE IF STATUS EQ 0 FROM PORT A.
014360 001414                      BEQ    71$                 ; BR IF ZERO
014362 013737 001224 001240      MOV    PORTA, PTNBR        ; SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014370 013737 001172 001126      MOV    $TMP3, $BDDAT       ; 'BAD DATA' FOR ERROR TYPE OUT
014376 113760 001224 000010      MOVB   PORTA, RMCS2(R0)    ; SELECT PORT A.
014404 005737 001166                      TST    $TMP1               ; SEE IF STATUS EQ ZERO FROM PORT B.
014410 001012                      BNE    72$                 ; BR IF NOT
014412 012737 177777 001254 71$:  MOV    #-1, RELERR        ; SET 'RELEASE ERROR' INDICATOR
014420 012760 000011 000000      MOV    #11, RMCS1(R0)     ; CLEAR THE DRIVE
  
```



```

014426 012760 000013 000000      MOV      #13,RMCS1(R0)      ;RELEASE THE DRIVE
014434 104026                      EMT      26
014436 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
014444 013737 001224 001240      MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
014452 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
014460 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;ALL BITS OK ?
014466 001401                      BEQ      73$              ;BR IF OK FROM PORT A.
014470 104007                      EMT      7
014472 013737 001172 001126 73$:  MOV      $TMP3,$BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
014500 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
014506 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
014514 023737 001124 001126      CMP      $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
014522 001401                      BEQ      74$              ;BR IF OK
014524 104007                      EMT      7
014526 000240                      NOP
014530 000004                      1$:      SCOPE              ;LOOP ?
  
```

328
347
348

```

:*****
:*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.
:*
:*****
  
```

```

014532
014532 005737 001300
014536 001406
014540 100002
014542 000137 003074
014546 012737 177777 001300 1$:
014554 012737 014570 001106 2$:
014562 012737 014570 001110
014570
  
```

```

TST11:
      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      #TEST11,$LPADR ;SETUP SCOPE LOOP ADDRESS
      MOV      #TEST11,$LPERR ;SETUP ERROR LOOP ADDRESS
  
```

1
2
3
4
5
6
7
8
9
10
11

.REM

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

IDENTIFICATION

PRODUCT CODE: AC-F936B-MC
PRODUCT NAME: CZRMRB0 RM05/3/2 DUAL PORT TEST, PT 1
PRODUCT DATE: APRIL 1981
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 UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1980,1981 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

- 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 OPERATOR ACTION
- 5. OPERATING PROCEDURES
 - 5.1 'SOFTWARE' SWITCH REGISTER
 - 5.2 OPERATIONAL SWITCH SETTINGS
 - 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
571. 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 PH70.

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
20K MEMORY
KW11-L OR KW11-P (CLOCK
TERMINAL
RH11 OR RH70
1 - DISK DRIVE (RM05, RM03 OR RM02)
RM DUAL PORT TEST CABLE (P/N: 7010507-02)

2.2 PREREQUISITE PROGRAMS

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

THE PRELIMINARY PROGRAMS MUST BE RUN TWICE: ONCE FROM EACH PORT (A & B).

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 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 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.

5. OPERATING PROCEDURES

5.1 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR 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 RM80 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

'SWR = NNNNNN NEW ='

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

EACH TIME SWITCH SETTINGS 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 1'S IN THE SWITCHES. ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

5.2 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

5.3 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.

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

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

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 RH11/RH70 TO J3 OF THE
RM05/3/2 BACK PANEL, THEN CONNECT THE TEST CABLE (P/N: 7010507-02)
FROM J2 TO J7 OF THE BACK PANEL AND TERMINATE THE PORT 'B' AT J6.

WHEN THE DUAL PORT TEST CABLE IS CONNECTED, THE ATTENTION
BITS FOR PORTS A & B ARE ASSEPTED 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

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

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

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 PORT OPTION, AND DOES NOT TEST THE UNLOAD COMMAND OR THE OPERATION OF THE PORT SELECT SWITCH ON THE DRIVE. (REFER TO PARAGRAPH 2.2 & 2.3)

7.3 EXECUTION TIME

PASS 1 OF THE PROGRAM TAKES ABOUT 25 SECONDS. PASS 2 AND SUBSEQUENT PASSES TAKE 2 MINUTES 25 SECONDS.

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

IF SW<09> IS SET, THE PROGRAM WILL LOOP ON A FAILING TEST UNTIL EITHER THE SWITCH IS RESET OR THE ERROR STOPS OCCURRING. 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 (RMD5) 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 RMD5, THE PROGRAM CONCLUDES THAT THE DRIVE IS IN NEUTRAL AND THAT ANY BIT DISCREPANCY BETWEEN PORTS INDICATES A

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

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 '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.

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

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 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.

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.

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.

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 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'.
- 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.

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

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 RMD5.
- 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 RMD5 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 CONTROLLER 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 RMD5 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 CONTROLLER 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

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

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.
- 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.

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

- A. WRITE THE APPROPRIATE DRIVE BIT INTO RMA5; 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 RMA5 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 RMD5.
- B. WRITE 0'S INTO RMD5 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.

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 RMD5.
- B. WRITE 0'S INTO RMD5 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

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

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.
- 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.

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

- 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 RESET.

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

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 RMA. VERIFY THAT BOTH ATTENTION BITS ARE RESET AS SEEN BY RMA.

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'.

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

- B. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD5.
- 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.

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.

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'.

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 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'.

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'.

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

- 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 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.

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

TEST 50 PORT 'A' SEIZE ACCESS TEST

VERIFY THAT THERE IS NO INTERACTION BETWEEN PORTS.

- A. SEIZE THE DRIVE THROUGH PORT 'A' BY WRITING 0'S INTO RMD5.
- 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 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.

1
597
598

```

;*LAST REVISION 04-APR-81
.TITLE CZRMRB0 RM05/3/2 DU POR TST 1
;*COPYRIGHT (C) 1981
;*DIGITAL EQUIPMENT CORPORATION
;*COLORADO SPGS., CO. 80919
;*
;*PROGRAM BY MIKE LEAVITT
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE FDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C5), 18-MAR-81

```

599

```

.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

```

600
601

```

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

```

001100
104000
000004

```

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

```

000011
000012
000015
000200
177776
177776
177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

```

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

```

000000
000040

```

;*PRIORITY LEVEL DEFINITIONS
PRO = 0 ;;PRIORITY LEVEL 0
PRI = 40 ;;PRIORITY LEVEL 1

```


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 RFSVEC = 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
    
```

602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639

.SBTTL RH/RM 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 JPE = 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

```

640          ;CONTROL AND STATUS REGISTER #1. (#00)
641
642          000001      GO      = 1          ;GO BIT (BIT #0)
643          000002      FO      = 2          ;FUNCTION CODE BIT #1
644          000004      F1      = 4          ;FUNCTION CODE BIT #2
645          000010      F2      = 10         ;FUNCTION CODE BIT #3
646          000020      F3      = 20         ;FUNCTION CODE BIT #4
647          000040      F4      = 40         ;FUNCTION CODE BIT #5
648          004000      DVA     = 4000       ;DEVICE AVAILABLE (BIT #11)
649
650          ;CONTROL STATUS REGISTER #2 (RMCS2)
651
652          000040      CLR     = BIT5       ;CONTROLLER CLEAR
653
654          ;DRIVE STATUS REGISTER (RMDS) (#01)
655
656          000001      OM      = BIT00      ;OFFSET MODE
657          000100      VV      = 100       ;VOLUME VALID (BIT #6)
658          000200      DRY     = 200       ;DRIVE READY (BIT #7)
659          000400      DPR     = 400       ;DRIVE PRESENT (BIT #8)
660          001000      PGM     = 1000      ;PROGRAMABLE (BIT #9)
661          002000      LBT     = 2000      ;LAST SECTOR TRANSFERRED (BIT #10)
662          004000      WRL     = 4000      ;WRITE LOCK (BIT #11)
663          010000      MOL     = 10000     ;MEDIUM ON-LINE (BIT #12)
664          020000      PIP     = 20000     ;POSITIONING OPERATION IN PROGRESS (BIT #13)
665          040000      ERR     = 40000     ;COMPOSITE ERROR (BIT #14)
666          100000      ATA     = 100000    ;ATTENTION ACTIVE (BIT #15)
667
668          ;ERROR REGISTER #01 (RMER1) (#02)
669
670          000001      ILF     = 1          ;ILLEGAL FUNCTION (BIT #0)
671          000002      ILR     = 2          ;ILLEGAL REGISTER (BIT #1)
672          000004      RMR     = 4          ;REGISTER MODIFICATION REFUSED (BIT #2)
673          000010      PAR     = 10         ;PARITY ERROR (BIT #3)
674          000020      FER     = 20         ;FORMAT ERROR (BIT #4)
675          000040      WCF     = 40         ;WRITE CLOCK FAIL (BIT #5)
676          000100      ECH     = 100       ;ECC HARD ERROR (BIT #6)
677          000200      HCE     = 200       ;HEADER COMPARE ERROR (BIT #7)
678          000400      HCRC    = 400       ;HEADER CRC ERROR (BIT #8)
679          001000      AOE     = 1000      ;ADDRESS OVERFLOW ERROR (BIT #9)
680          002000      IAE     = 2000      ;INVALID ADDRESS ERROR (BIT #10)
681          004000      WLE     = 4000      ;WRITE LOCK ERROR (BIT #11)
682          010000      DTE     = 10000     ;DRIVE TIMING ERROR (BIT #12)
683          020000      OPI     = 20000     ;OPERATION INCOMPLETE (BIT #13)
684          040000      UNS     = 40000     ;DRIVE UNSAFE (BIT #14)
685          100000      DCK     = 100000    ;DATA CHECK ERROR (BIT 15)
686
687          ;MAINTAINABILITY REGISTER (RMMR1) (#03)
688
689          000001      DMD     = 1          ;DIAGINOSTIC MODE (BIT #0)
690          001000      MUR     = BIT09     ;MAINTENANCE UNIT READY
691          040000      RQB     = BIT14     ;PORT B REQUEST FLOP
692          100000      RQA     = BIT15     ;PORT A REQUEST FLOP
693
694          ;ATTENTION SUMMARY PSEUDO-REGISTER (RMAS) (#04)
695
696          000001      ATO     = 1          ;DEVICE 0 (BIT #0)
  
```

697	000002	AT1	= 2	:DEVICE 1 (BIT #1)
698	000004	AT2	= 4	:DEVICE 2 (BIT #2)
699	000010	AT3	= 10	:DEVICE 3 (BIT #3)
700	000020	AT4	= 20	:DEVICE 4 (BIT #4)
701	000040	AT5	= 40	:DEVICE 5 (BIT #5)
702	000100	AT6	= 100	:DEVICE 6 (BIT #6)
703	000200	AT7	= 200	:DEVICE 7 (BIT #7)
704				
705		:DESIRED SECTOR/TRACK ADDRESS REGISTER (RMDA) (#05)		
706		:(EACH BIT IS CALLED BY BIT NUMBER)		
707				
708		:DRIVE TYPE REGISTER (RMDT) (#06)		
709				
710	000001	DT00	= 1	:DRIVE TYPE NUMBER BIT 1
711	000002	DT01	= 2	:DRIVE TYPE NUMBER BIT 2
712	000004	DT02	= 4	:DRIVE TYPE NUMBER BIT 3
713	000010	DT03	= 10	:DRIVE TYPE NUMBER BIT 4
714	000020	DT04	= 20	:DRIVE TYPE NUMBER BIT 5
715	000040	DT05	= 40	:DRIVE TYPE NUMBER BIT 6
716	000100	DT06	= 100	:DRIVE TYPE NUMBER BIT 7
717	000200	DT07	= 200	:DRIVE TYPE NUMBER BIT 8
718	000400	DT08	= 400	:DRIVE TYPE NUMBER BIT 9
719	004000	DRQ	= 4000	:DRIVE REQUEST REQUIRED (BIT #11)
720	020000	MOH	= 20000	:MOVING HEAD (BIT #13)
721	040000	TAP	= 40000	:TAPE DRIVE (BIT #14)
722	100000	NBA	= 100000	:NOT BLOCK ADDRESSED (BIT #15)
723				
724		:LOOK-AHEAD REGISTER (RMLA) (#07)		
725				
726	000100	SC0	= 100	:SECTOR COUNT FIELD 0 (BIT #6)
727	000200	SC1	= 200	:SECTOR COUNT FIELD 1 (BIT #7)
728	000400	SC2	= 400	:SECTOR COUNT FIELD 2 (BIT #8)
729	001000	SC3	= 1000	:SECTOR COUNT FIELD 3 (BIT #9)
730	002000	SC4	= 2000	:SECTOR COUNT FIELD 4 (BIT #10)
731				
732		:RM ERROR REGISTER #2 (RMER2) (#10)		
733				
734	000010	DPE	= 10	:DATA PARITY ERROR (BIT #3)
735	000200	DVC	= 200	:DEVICE CHECK (BIT #7)
736	002000	LBC	= 2000	:LOSS OF BIT CLOCK (BIT #10)
737	004000	LSC	= 4000	:LOSS OF SYSTEM CLOCK (BIT #11)
738	010000	IVC	= 10000	:INVALID COMMAND (BIT #12)
739	020000	GPE	= 20000	:OPERATOR ERROR (BIT #13)
740	100000	SKI	= 100000	:SEEK INCOMPLETE (BIT #14)
741				
742		:OFFSET REGISTER (RMOF) (#11)		
743				
744	000200	OFD	= 200	:OFFSET FORWARD (BIT #5)
745	002000	HCI	= 2000	:HEADER COMPARE INHIBIT (BIT #10)
746	004000	ECI	= 4000	:ERROR CORRECTION CODE INHIBIT (BIT #11)
747	010000	FMT16	= 10000	:FORMAT BIT (BIT #12)
748				
749		:DESIRED CYLINDER ADDRESS (RMDC) (#12)		
750		:(EACH BIT IS CALLED BY BIT NUMBER)		
751				
752		:SERIAL NUMBER REGISTER (RMSN) (#14)		
753		:(EACH IS CALLED BY BIT NUMBER)		

```

754
755      ;ECC POSITION REGISTER (RMEC1) (#16)
756      ;(EACH BIT IS CALLED BY BIT NUMBER)
757
758      ;ECC PATTERN REGISTER (RMEC2) (#17)
759      ;(EACH BIT IS CALLED BY BIT NUMBER)
760
761      .SBTTL  DEFINITIONS OF THE RH/RM ADDRESS INDEXES
762
763      000000      RMCS1   = 0      ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
764      000002      RMWC    = 2      ;WORD COUNT REGISTER (NOT A DRIVE REG)
765      000004      RMBA    = 4      ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
766      000006      RMDA    = 6      ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
767      000010      RMCS2   = 10     ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
768      000012      RMDS    = 12     ;DRIVE STATUS REGISTER (DRIVE REG 01)
769      000014      RMER1   = 14     ;ERROR REGISTER #1 (DRIVE REG. 02)
770      000016      RMAS    = 16     ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
771      000020      RMLA    = 20     ;LOOK AHEAD REGISTER (DRIVE REG. 07)
772      000022      RMDB    = 22     ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
773      000024      RMMR1   = 24     ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
774      000026      RMDT    = 26     ;DRIVE TYPE REGISTER (DRIVE REG. 06)
775      000030      RMSN    = 30     ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
776      000032      RMOF    = 32     ;OFFSET REGISTER (DRIVE REG. 11)
777      000034      RMDC    = 34     ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
778      000040      RMMR2   = 40     ;MAINTENANCE REGISTER #2 (DRIVE REG. 14)
779      000042      RMER2   = 42     ;ERROR REGISTER #2 (DRIVE REG. 15)
780      000044      RMEC1   = 44     ;ECC POSITION REGISTER (DRIVE REG. 16)
781      000046      RMEC2   = 46     ;ECC PATTERN REGISTER (DRIVE REG. 17)
782
  
```

```

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

.SBTTL STARTING ADDRESS(ES)

2 000200      000137      002240      JMP      @#START      ;;JUMP TO STARTING ADDRESS OF PROGRAM
3 000204      000137      002246      JMP      @#START1     ;START AND CHANGE THE RH/RM ADDRESS
4
5

.SBTTL ACT11 HOOKS

:*****
:HOOKS REQUIRED BY ACT11
000210      000210      $SVPC=.      ;SAVE PC
000046      000046      .=46
000046      066110      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
000052      000052      .=52
000052      020000      .WORD 20000      ;;2)SET LOC.52 TO 20000
6 000210      000210      .= $SVPC      ;; RESTORE PC

```

0

.SBTTL COMMON TAGS

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

001100	001100			SCMTAG: .WORD	0	:: START OF COMMON TAGS
001100	000000			\$PASS: .WORD	0	:: CONTAINS PASS COUNT
001102	000			\$TSTNM: .BYTE	0	:: CONTAINS THE TEST NUMBER
001103	000			\$ERFLG: .BYTE	0	:: CONTAINS ERROR FLAG
001104	000000			\$ICNT: .WORD	0	:: CONTAINS SUBTEST ITERATION COUNT
001106	000000			\$LPADR: .WORD	0	:: CONTAINS SCOPE LOOP ADDRESS
001110	000000			\$LPERR: .WORD	0	:: CONTAINS SCOPE RETURN FOR ERRORS
001112	000000			\$ERTTL: .WORD	0	:: CONTAINS TOTAL ERRORS DETECTED
001114	000			\$ITEMB: .BYTE	0	:: CONTAINS ITEM CONTROL BYTE
001115	001			\$ERMAX: .BYTE	1	:: CONTAINS MAX. ERRORS PER TEST
001116	000000			\$ERRPC: .WORD	0	:: CONTAINS PC OF LAST ERROR INSTRUCTION
001120	000000			\$GDADR: .WORD	0	:: CONTAINS ADDRESS OF 'GOOD' DATA
001122	000000			\$BDADR: .WORD	0	:: CONTAINS ADDRESS OF 'BAD' DATA
001124	000000			\$GDDAT: .WORD	0	:: CONTAINS 'GOOD' DATA
001126	000000			\$BDDAT: .WORD	0	:: CONTAINS 'BAD' DATA
001130	000C00				0	:: RESERVED--NOT TO BE USED
001132	000000				0	
001134	000			\$AUTOB: .BYTE	0	:: AUTOMATIC MODE INDICATOR
001135	000			\$INTAG: .BYTE	0	:: INTERRUPT MODE INDICATOR
001136	000000				0	
001140	177570			\$SWR: .WORD	DSWR	:: ADDRESS OF SWITCH REGISTER
001142	177570			\$DISPLAY: .WORD	DDISP	:: ADDRESS OF DISPLAY REGISTER
001144	177560			\$TKS: .WORD	177560	:: TTY KBD STATUS
001146	177562			\$TKB: .WORD	177562	:: TTY KBD BUFFER
001150	177564			\$TPS: .WORD	177564	:: TTY PRINTER STATUS REG. ADDRESS
001152	177566			\$TPB: .WORD	177566	:: TTY PRINTER BUFFER REG. ADDRESS
001154	000			\$NULL: .BYTE	0	:: CONTAINS NULL CHARACTER FOR FILLS
001155	002			\$FILLS: .BYTE	2	:: CONTAINS # OF FILLER CHARACTERS REQUIRED
001156	012			\$FILLC: .BYTE	12	:: INSERT FILL CHARS. AFTER A 'LINE FEED'
001157	000			\$TPFLG: .BYTE	0	:: 'TERMINAL AVAILABLE' FLAG (BIT<07>=0-YES)
001160	000000			\$REGAD: .WORD	0	:: CONTAINS THE ADDRESS FROM WHICH (\$REGO) WAS OBTAINED
001162	000000			\$REGO: .WORD	0	:: CONTAINS ((\$REGAD)+0)
001164	000000			\$TMP0: .WORD	0	:: USER DEFINED
001166	000000			\$TMP1: .WORD	0	:: USER DEFINED
001170	000000			\$TMP2: .WORD	0	:: USER DEFINED
001172	000000			\$TMP3: .WORD	0	:: USER DEFINED
001174	000000			\$TMP4: .WORD	0	:: USER DEFINED
001176	000000			\$TIMES: .WORD	0	:: MAX. NUMBER OF ITERATIONS
001200	000000			\$ESCAPE: .WORD	0	:: ESCAPE ON ERROR ADDRESS
001202	207	377	377	\$BELL: .ASCIZ	<207><377><377>	:: CODE FOR BELL
001206	077			\$QUES: .ASCII	/?/	:: QUESTION MARK
001207	015			\$CRLF: .ASCII	<15>	:: carriage RETURN
001210	012	000		\$LF: .ASCIZ	<12>	:: LINE FEED

.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	ASR†: .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

.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				
2					
3					
4	001310	072724	EM1	:WRONG DRIVE TYPE	
5	001312	077341	DH1		
6	001314	101230	DT1		
7	001316	101516	DF1		
8					
9					
10					
11	001320	072745	EM2	:DRIVE NOT ON LINE	
12	001322	077341	DH1		
13	001324	101230	DT1		
14	001326	101516	DF1		
15					
16					
17					
18	001330	072767	EM3	:SERIAL NUMBERS NOT THE SAME	
19	001332	077412	DH3		
20	001334	101244	DT3		
21	001336	101516	DF1		
22					
23					
24					
25	001340	073051	EM4	:DRIVE NOT SEIZED BY PORT 'N'	
26	001342	077461	DH4		
27	001344	101312	DT7		
28	001346	101531	DF7		
29					
30					
31					
32	001350	073102	EM5	:WRONG STATUS SEEN BY THE SEIZING PORT	
33	001352	077604	DH5		
34	001354	101260	DT5		
35	001356	101523	DF5		
36					
37					
38					
39	001360	073150	EM6	:REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED	
40	001362	100054	DH13		
41	001364	101332	DT13		
42	001366	101523	DF5		

43				
44			:ERROR 7	
45				
46	001370	073250	EM7	:REGISTER CONTENTS INCORRECT AFTER RELEASE/TIMEOUT
47	001372	077660	DH7	
48	001374	101312	DT7	
49	001376	101531	DF7	
50				
51			:ERROR 10	
52				
53	001400	073331	EM10	:REGISTER CONTENTS INCORRECT
54	001402	077604	DH5	
55	001404	101260	DT5	
56	001406	101523	DF5	
57				
58			:ERROR 11	
59				
60	001410	073361	EM11	:CONTROL BUS PARITY ERROR WHILE READING REGISTER
61	001412	100003	DH11	
62	001414	101230	DT1	
63	001416	101516	DF1	
64				
65			:ERROR 12	
66				
67	001420	073445	EM12	:DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND
68	001422	100550	DH36	
69	001424	101420	DT37	
70	001426	101544	DF36	
71				
72			:ERROR 13	
73				
74	001430	073515	EM13	: 'VOLUME VALID' BIT NOT SET BY READIN PRESET
75	001432	100054	DH13	
76	001434	101332	DT13	
77	001436	101523	DF5	
78				
79			:ERROR 14	
80				
81	001440	073602	EM14	: 'VOLUME VALID' SET ON THE OPPOSITE PORT
82	001442	100054	DH13	
83	001444	101332	DT13	
84	001446	101523	DF5	
85				
86			:ERROR 15	
87				
88	001450	073645	EM15	:THE ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET
89	001452	077660	DH7	
90	001454	101312	DT7	
91	001456	101531	DF7	
92				
93			:ERROR 16	
94				
95	001460	073724	EM16	:ATTN BIT WRONG AFTER RELEASE - REQUEST WAS SET
96	001462	077660	DH7	
97	001464	101312	DT7	
98	001466	101531	DF7	
99				

100			:ERROR 17	
101				
102	001470	073777	EM17	:ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET
103	001472	077660	DH7	
104	001474	101312	DT7	
105	001476	101531	DF7	
106				
107			:ERROR 20	
108				
109	001500	074056	FM20	:DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED
110	001502	100550	DH36	
111	001504	101420	DT37	
112	001506	101544	DF36	
113				
114			:ERROR 21	
115				
116	001510	074136	EM21	:DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT FOR PORT
117	001512	100550	DH36	
118	001514	101420	DT37	
119	001516	101544	DF36	
120				
121			:ERROR 22	
122				
123	001520	074211	EM22	:DRIVE NOT IN NEUTRAL AFTER TIMEOUT, REQUEST NOT SET
124	001522	100174	DH22	
125	001524	101350	DT22	
126	001526	101540	DF31	
127				
128			:ERROR 23	
129				
130	001530	074276	EM23	:TIMEOUT CLEARED THE DRIVE'S ERROR BIT
131	001532	100272	DH23	
132	001534	101362	DT23	
133	001536	101516	DF1	
134				
135			:ERROR 24	
136				
137	001540	074344	EM24	:RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET
138	001542	100272	DH23	
139	001544	101362	DT23	
140	001546	101516	DF1	
141				
142				
143			:ERROR 25	
144				
145	001550	074423	EM25	:TIMEOUT ONE-SHOT DID NOT RETRIGGER
146	001552	100550	DH36	
147	001554	101410	DT36	
148	001556	101544	DF36	
149				
150				
151			:ERROR 26	
152				
153	001560	074466	EM26	:DRIVE NOT IN NEUTRAL AFTER RELEASE, REQUEST NOT SET
154	001562	100174	DH22	
155	001564	101350	DT22	
156	001566	101540	DF31	

157				
158				
159				
160	001570	074553	EM27	;REGISTER WRONG AFTER RELEASE WITH REQUEST SET
161	001572	077660	DH7	
162	001574	101312	DT7	
163	001576	101531	DF7	
164				
165				
166				
167	001600	074631	FM30	;DRIVE SEIZED BY RELEASE ISSUED WHEN DRIVE IN NEUTRAL
168	001602	100550	DH36	
169	001604	101410	DT36	
170	001606	101544	DF36	
171				
172				
173				
174	001610	074726	EM31	;DRIVE NOT SEIZED BY PORT AFTER RELEASE WITH REQUEST SET
175	001612	100451	DH31	
176	001614	101376	DT31	
177	001616	101540	DF31	
178				
179				
180				
181	001620	075003	EM32	;ATTN BIT WRONG AFTER RECALIBRATE COMMAND
182	001622	077604	DH5	
183	001624	101260	DT5	
184	001626	101523	DF5	
185				
186				
187				
188	001630	075054	EM33	;DRIVE RETURNS TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED
189	001632	100550	DH36	
190	001634	101410	DT36	
191	001636	101544	DF36	
192				
193				
194				
195	001640	075156	EM34	;DRIVE RETURNS TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED
196	001642	100550	DH36	
197	001644	101410	DT36	
198	001646	101544	DF36	
199				
200				
201				
202	001650	075261	EM35	;DRIVE DID NOT RETURN TO NEUTRAL BY TRIGGERING TIMEOUT ONE SHOT
203	001652	100550	DH36	
204	001654	101420	DT37	
205	001656	101544	DF36	
206				
207				
208				
209	001660	075340	EM36	;TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS
210	001662	100550	DH36	
211	001664	101410	DT36	
212	001666	101544	DF36	
213				

214			:ERROR 37	
215				
216	001670	075412	EM37	:DRIVE IS NON-EXISTENT
217	001672	100550	DH36	
218	001674	101420	DT37	
219	001676	101544	DF36	
220				
221			:ERROR 40	
222				
223	001700	075460	EM40	:ATTENTION FOR PORT NOT RESET BY MASSBUS CLEAR
224	001702	077341	DH1	
225	001704	101362	DT23	
226	001706	101516	DF1	
227				
228			:ERROR 41	
229				
230	001710	075535	EM41	:TIMEOUT CLEARED ATTENTION BIT
231	001712	100272	DH23	
232	001714	101362	DT23	
233	001716	101516	DF1	
234				
235			:ERROR 42	
236				
237	001720	075577	EM42	:DRIVE NOT IN NEUTRAL OR SEIZED
238	001722	100577	DH42	
239	001724	101430	DT2	
240	001726	101547	DF42	
241				
242			:ERROR 43	
243				
244	001730	075665	EM43	:DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN
245	001732	100577	DH42	
246	001734	101430	DT42	
247	001736	101547	DF42	
248				
249			:ERROR 44	
250				
251	001740	075742	EM44	:WRITE ATTENTION BIT DID NOT SET PORT REQUEST
252	001742	100616	DH44	
253	001744	101376	DT31	
254	001746	101540	DF31	
255				
256			:ERROR 45	
257				
258	001750	076017	EM45	:PORT SELECT SWITCH ON DRIVE NOT IN 'A/B'
259	001752	077341	DH1	
260	001754	101230	DT1	
261	001756	101516	DF1	
262				
263			:ERROR 46	
264				
265	001760	076071	EM46	:CAN'T ACCESS DRIVE THROUGH EITHER PORT
266	001762	100714	DH46	
267	001764	101436	DT46	
268	001766	101540	DF31	
269				
270			:ERROR 47	

ERROR POINTER TABLE

271				
272	001770	076140	EM47	:ATTN BIT FOR SEIZING PORT NOT CLEARED BY DRIVE CLEAR
273	001772	100272	DH23	
274	001774	101362	DT23	
275	001776	101516	DF1	
276				
277				:ERROR 50
278				
279	02000	076226	EM50	:ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR COMMAND
280	002002	100054	DH13	
281	002004	101332	DT13	
282	002006	101523	DF5	
283				
284				:ERROR 51
285				
286	002010	076310	EM51	:ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL
287	002012	077604	DH5	
288	002014	101260	DT5	
289	002016	101523	DF5	
290				
291				:ERROR 52
292				
293	002020	076377	EM52	:ATTN BIT SET AFTER TIMEOUT, 'ERR' SET, NO REQUEST
294	002022	100054	DH13	
295	002024	101332	DT13	
296	002026	101523	DF5	
297				
298				:ERROR 53
299				
300	002030	076472	EM53	:CAN'T READ ATTN BIT FROM OPPOSITE PORT
301	002032	100272	DH23	
302	002034	101230	DT1	
303	002036	101516	DF1	
304				
305				:ERROR 54
306				
307	002040	076553	EM54	:RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT
308	002042	100174	DH22	
309	002044	101450	DT54	
310	002046	101540	DF31	
311				
312				:ERROR 55
313				
314	002050	076646	EM55	:TIMEOUT ONE-SHOT IS LESS THAN 500 MS
315	002052	101011	DH55	
316	002054	101462	DT55	
317	002056	101551	DF55	
318				
319				:ERROR 56
320				
321	002060	076713	EM56	:RH/RM DIDN'T RESPOND TO ADDRESSING
322	002062	101067	DH56	
323	002064	101474	DT56	
324	002066	101555	DF56	
325				
326				
327				:ERROR 57

328				
329	002070	076756	EM57	;PORT REQUEST FLOPS WRONG
330	002072	101076	DH57	
331	002074	101500	DT57	
332	002076	101523	DF5	
333				
334				;ERROR 60
335				
336	002100	077017	EM60	;ATTENTION BITS NOT RESET BY RMAS
337	002102	077604	DH5	
338	002104	101260	DT5	
339	002106	101523	DF5	
340				
341				;ERROR 61
342				
343	002110	077063	EM61	;ATTENTION NOT RESET BY GO
344	002112	100272	DH23	
345	002114	101362	DT23	
346	002116	101516	DF1	
347				
348				;ERROR 62
349				
350	002120	077115	EM62	;ATTENTION RESET BY GO WHEN NOT SEIZED
351	002122	100054	DH13	
352	002124	101332	DT13	
353	002126	101523	DF5	
354				
355				;ERROR 63
356				
357	002130	077163	EM63	;DRIVE SEIZED BY UNIT READY CHANGE
358	002132	100550	DH36	
359	002134	101410	DT36	
360	002136	101544	DF36	
361				
362				;ERROR 64
363				
364	002140	077225	EM64	;ATTENTION NOT SET BY UNIT READY CHANGE
365	002142	077660	DH7	
366	002144	101312	DT7	
367	002146	101531	DF7	
368				
369				;ERROR 65
370				
371	002150	077274	EM65	;VV NOT RESET BY UNIT READY
372	002152	077604	DH5	
373	002154	101260	DT5	
374	002156	101523	DF5	
375				


```

1      ;THIS ROUTINE HANDLES UNEXPECTED TIMEOUTS
2
3      002160 011600      BADTMO: MOV    (SP),R0      ;SAVE PC WHERE THE TIME OUT OCCURED
4      002162 005740      TST    -(R0)      ;ADJUST PC -2
5      002164 022626      CMP    (SP)+,(SP)+    ;RESTORE STACK POINTER
6      002166 104401 002174  TYPE    ,65$      ;:TYPE ASCIZ STRING
        002172 000417      BR     64$      ;:GET OVER THE ASCIZ
        ;:65$: .ASCIZ <CRLF>/UNEXPECTED BUS TIMEOUT, PC=/
        64$:
7      002232 010046      MOV    R0,-(SP)      ;SETUP FOR TYPING OUT PC
8      002234 104402      TYPOC
9      002236 000240      NOP
        ;PUT 'HALT(0)' INSTRUCTION HERE IF YOU WISH
        ;TO STOP ON UNEXPECTED TIMEOUT.
10
11
12      .SBTTL  START OF PROGRAM
13
14      002240 005037 001302  START: CLR    CHGADR      ;CLEAR THE 'CHANGE RH/RM ADDRESS' INDICATOR
15      002244 000403      BR     START2      ;GO TO THE START
16
17      002246 012737 177777 001302  START1: MOV   #-1,CHGADR ;SET THE 'CHANGE RH/RM ADDRESS' INDICATOR
18
19      002254 000240      START2: NOP
20      002256 005227 000000      INC    #0          ;TTY LOOP, WAIT FOR INCREMENT
21      002262 001375      BNE   #-4          ;OF WORD
22      002264 000005      RESET
        ;CLEAR THE WORLD
23
24      .SBTTL  INITIALIZE THE COMMON TAGS
        ;:CLEAR THE COMMON TAGS ($CMTAG) AREA
        MOV   #SCMTAG,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   $ESCAPE       ;:CLEAR THE ESCAPE ON FRROR ADDRESS
        MOVB  #1,$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 RTI
002460 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
002466 012737 000174 001142 MOV #DISPREG,DISPLAY
002474 012637 000004 66$: 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
29 .SBTTL TYPE PROGRAM NAME
;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
002514 005227 177777 INC #-1 ;;FIRST TIME?
002520 001037 BNE 67$ ;;BRANCH IF NO
002522 022737 066110 000042 CMP #SENDAD,@#42 ;;ACT-11?
002530 001433 BEQ 67$ ;;BRANCH IF YES
002532 104401 002540 TYPE ,68$ ;;TYPE ASCIZ STRING
002536 000430 BR 67$ ;;GET OVER THE ASCIZ
;;68$: .ASCIZ <CRLF>@CZMRBO - RM05/3/2 DUAL PORT LOGIC TEST, PT 1@<CRLF>
67$:
.SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
002620 005737 000042 TST @#42 ;;ARE WE RUNNING UNDER XXDP/ACT?
002624 001006 BNE 69$ ;;BRANCH IF YES
002626 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
002634 001005 BNE 70$ ;;BRANCH IF NO
002636 104406 GTSWR ;;GET SOFT-SWR SETTINGS
002640 000403 BR 70$
002642 112737 000001 001134 69$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
70$:

50
31 002650 004737 070610 JSR PC,$TKINT ;;SETUP THE TTY KEYBOARD
32 002654 004737 003246 1$: JSR PC,CHANGE ;;CHECK/CHANGE THE RH/RM ADDRFS
33 002660 104401 072372 TYPE ,ENTERA ;;ENTER DRIVE ADDRESS
34 002664 104412 RDOCT ;;GET THE ADDRESS
35 002666 012637 001224 MOV (SP)+,PORTA ;;STORE THE ADDRESS
36 002672 023727 001224 000007 CMP PORTA,#7 ;;SEE IF ADDRESS TOO LARGE
37 002700 101403 BLOS 2$ ;;BR IF NOT
38 002702 104401 072421 TYPE ,ADRERR ;;TYPE ADDRESS ERROR MESSAGE
39 002706 000762 BR 1$ ;;TRY AGAIN
40 002710 013737 001224 001226 2$: MOV PORTA,PORTB ;;GENERATE THE PORT B ADDRESS
41 002716 005237 001226 INC PORTB ;;INCREMENT THE ADDRESS
42 002722 042737 000016 001226 BIC #16,PORTB ;;LEAVE BIT 0
43 002730 013746 001224 MOV PORTA,-(SP) ;;PUT PORT A ADDRESS ON THE STACK
44 002734 042716 177771 BIC #^C6,(SP) ;;SAVE BITS 1 & 2
45 002740 052637 001226 BIS (SP)+,PORTB ;;SET BITS 1 & 2 IN PORT B ADDRESS
46 002744 104401 072444 TYPE ,PORTAIS ;;PORT A ADDRESS IS '
47 002750 013746 001224 MOV PORTA,-(SP) ;;SAVE PORTA FOR TYPEOUT
;;TYPE PORT A ADDRESS
;;GO TYPE--OCTAL ASCII
002754 104403 TYPOS ;;TYPE 1 DIGIT(S)
002756 001 .BYTE 1 ;;SUPPRESS LEADING ZEROS
002757 000 .BYTE 0 ;;PORT B ADDRESS IS '
48 002760 104401 072473 TYPE ,PORTBIS ;;SAVE PORTB FOR TYPEOUT
49 002764 013746 001226 MOV PORTB,-(SP) ;;TYPE PORT B ADDRESS
;;GO TYPE--OCTAL ASCII
002770 104403 TYPOS ;;TYPE 1 DIGIT(S)
002772 001 .BYTE 1 ;;SUPPRESS LEADING ZEROS
002773 000 .BYTE 0 ;;ANOTHER CR-LF
50 002774 104401 001207 TYPE ,SCRLF

```

```

51 003000 013737 001224 001230      MOV     PORTA,PORTC      ;GENERATE ADDRESS OF DRIVE NOT TESTED
52 003006 C62737 000006 001230      ADD     #6,PORTC        ;COMPLEMENT SOME BITS
53 003014 042737 177770 001230      BIC     #^C7,PORTC     ;SAVE ONLY LOWER BITS
54 003022 013701 001224          MOV     PORTA,R1        ;USE PORT A ADDRESS AS INDEX
55 003026 116137 101672 001236      MOVVB  ATABIT(R1),ASR1  ;GET ATTENTION BIT FOR DRIVE
58 003034 005037 001262          CLR     TIMEA           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003040 005037 001264          CLR     TIMEAP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003044 005037 001270          CLR     TIMEB           ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
   003050 005037 001272          CLR     TIMEBP          ;CLEAR TIMEOUT ONE-SHOT VALUE LOCATION
59 003054 004737 066130      JSR     PC,CKCLK        ;SETUP CLOCK
60 003060 000137 003074      JMP     EXEC            ;CLOCK HAS BEEN STARTED
61 003064 104401 072522          TYPE   ,NOCLOCK        ;NO CLOCK ON SYSTEM
62 003070 000000          HALT   3$             ;FATAL ERROR
63 003072 000776          BR     3$             ;INTERLOCK THE HALT
64
65                                     ;ROUTINE TO GET THE TEST NUMBER FROM THE OPERATOR
66
67 003074 000005          EXEC:  RESET          ;CLEAR EVERYTHING
68 003076 005037 177776          CLR     PS              ;CLEAR THE PROCESSOR STATUS WORD
69 003102 104401 001207          TYPE   ,%CRLF          ;CR-LF
70 003106 013700 001304          MOV     $RMADR,RO      ;RH/RM ADDRESS FOR INDEXING
71 003112 012706 001100          MOV     #STACK,SP     ;LOAD STACK POINTER
72 003116 004737 066130      JSR     PC,CKCLK        ;START THE CLOCK
73 003122 000240          NOP                    ;RETURN IF NO CLOCK
74 003124 004737 070610      JSR     PC,$TKINT      ;INITIALIZE THE KEYBOARD
75 003130 005037 001300          CLR     KYBCTL         ;CLEAR SINGLE TEST INDICATOR
76 003134 005037 001100          CLR     $PASS         ;CLEAR THE PASS COUNT
77 003140 112737 000001 001115      MOVVB  #1,$ERMAX      ;SET ERROR MAX TO 1
78 003146 012737 003146 001106      MOV     #,$SLPADR     ;INITIAL SETTING FOR LOOP ADDRESS
79 003154 012737 003154 001110      MOV     #,$SLPERR     ;INITIAL SETTING FOR LOOP ON ERROR ADDRESS
80 003162 104401 072560          1$:   TYPE   ,TESTNO  ;ASK FOR TEST NUMBER
81 003166 104412          RDOCT                    ;GET THE NUMBER
82 003170 012601          MOV     (SP)+,R1      ;PUT ENTRY INTO R1
83 003172 001002          BNE    2$             ;BR IF NOT ZERO
84 003174 000* 003374          JMP     TST1          ;ENTER ZERO - PERFORM ALL TESTS
85 003200 020,57 101702          2$:   CMP     R1,MAXTN  ;SEE IF NUMBER GREATER THAN MAXIMUM
86 003204 003403          BLE   3$             ;BR IF LESS OR EQUAL
87 003206 104401 072600          TYPE   ,BADNO        ;BAD ENTRY
88 003212 000763          BR     1$             ;TRY AGAIN
89 003214 005301          3$:   DEC     R1        ;DECREMENT ENTRY
90 003216 006301          ASL    R1             ;SHIFT IT LEFT
91 003220 016137 101556 003244      MOV     TSTADR(R1),4$ ;GET THE TEST ADDRESS
92 003226 005237 001300          INC    KYBCTL         ;SET SINGLE TEST INDICATOR
93 003232 012737 000001 001104      MOV     #1,$ICNT     ;PRESET ITERATION COUNT
94 003240 000177 000000          JMP     @4$          ;GO TO THE SELECTED TEST
95 003244 000000          4$:   .WORD  0        ;TEST ADDRESS GOES HERE
96
97                                     ;CHANGE THE RH/RM UNIBUS ADDRESS USED BY THE PROGRAM
98
99 003246 005737 001302          CHANGE: TST    CHGADR  ;CHANGE THE ADDRESS ?
100 003252 001421          BEQ    3$             ;BR IF NOT
101 003254 005037 001302          CLR    CHGADR        ;CLEAR THE INDICATOR
102 003260 104401 072640          1$:   TYPE   ,ADDRIS  ;TYPE OUT WHAT THE PRESENT ADDRESS IS
103 003264 013746 001304          MOV     $RMADR,-(SP)  ;PUT THE ADDRESS ON THE STACK
104 003270 104402          TYPOC                    ;TYPE THE ACTUAL ADDRESS
105 003272 104401 001207          TYPE   ,%CRLF        ;CR-LF
106 003276 104401 072675          TYPE   ,NTRH        ;ASK FOR NEW ADDRESS
    
```

107	003302	104412				RDOCT		
108	003304	005716				TST	(SP)	:0 OR 'CR' ENTERED ?
109	003306	001402				BEQ	2\$:BR IF EITHER ENTERED (NO ADDRESS CHANGE)
110	003310	011637	001304			MOV	(SP), \$RMADR	:NEW RH/RM ADDRESS
111	003314	005726			2\$:	TST	(SP)+	:CORRECT THE STACK POINTER
112	003316	012737	003336	000004	3\$:	MOV	#4\$, @#4	:LOAD TRAP ADDRESS
113	003324	013700	001304			MOV	\$RMADR, R0	:GET RH/RM ADDRESS
114	003330	005760	000002			TST	RMWC(R0)	:RESPONDS AT THAT ADDRESS ?
115	003334	000404				BR	5\$:BR IF YES
116	003336				4\$:			
	003336	104056				EMT	56	
117	003340	062706	000004			ADD	#4, SP	:RESET THE STACK POINTER
118	003344	000745				BR	1\$:GET ADDRESS AGAIN
119	003346	012737	000006	000004	5\$:	MOV	#6, @#4	:RESTORE THE VECTOR
120	003354	000207				RTS	PC	:RETURN

TESTS
 1
 2
 16
 17 003356 013700 001304
 18 003362 012746 000240
 003366 012746 003374
 003372 000002
 003374
 19
 20

.SBTTL TESTS

TST1AA: MOV \$RMADR,RO ;:RESTORE R0 AFTER END OF PASS
 MOV #PR5,-(SP) ;:PUT NEW PS ON STACK
 MOV #64\$,-(SP) ;:PUT NEW PC ON STACK
 RTI ;:POP NEW PC AND PS

64\$:

 *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, 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.

TST1:

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 #-T,KYBCTL ;:SET SINGLE TEST INDICATOR
 2\$: MOV #TEST1,\$LPADR ;:SETUP SCOPE LOOP ADDRESS
 MOV #TEST1,\$LPERR ;:SETUP ERROR LOOP ADDRESS

TEST1:

MOVB #1,\$STSTM ;:MOVE #1 TO TEST NUMBER
 MOV #STACK,SP ;:LOAD THE STACK POINTER
 MOV #1,\$TIMES ;:DO 1 ITERATION

MOV #CLR,RMCS2(R0) ;:INITIALIZE THE MASSBUS

;VERIFY THAT DRIVE IS PRESENT THROUGH PORTS A & B

MOVB PORTA,RMCS2(R0) ;:SELECT PORT A
 MOV PORTA,PTNBR ;:MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 TST RMDS(R0) ;:SEE IF DRIVE (PORT A) PRESENT
 CLR CKERR ;:CLEAR THE 'CHECK ERROR' INDICATOR
 MOV RMCS2(R0),\$BDDAT ;:GET CONTENTS OF RMCS2
 MOV #RMCS2,\$BDADR ;:FORM REGISTER ADDRESS OF ERROR MESSAGE
 ADD R0,\$BDADR ;:ADD RH/RM BASE ADDRESS
 CLR \$GDDAT ;:WHAT REGISTER SHOULD BE
 MOV \$BDDAT,\$TMP0 ;:MOVE REGISTER CONTENTS TO '\$TMP0'
 BIC #^CNED,\$TMP0 ;:SAVE SPECIFIED BITS
 CMP \$GDDAT,\$TMP0 ;:COMPARE THE BITS
 BEQ 64\$;:BR IF OK
 MOV \$BDDAT,\$TMP4 ;:COPY 'BAD DATA'
 BIC #NED,\$TMP4 ;:CLEAR THE MASKED BITS
 BIS \$TMP4,\$GDDAT ;:'OR' WITH GOOD DATA FOR TYPEOUT
 EMT 37
 COM CKERR ;:SET THE REGISTER COMPARE ERROR INDICATOR

003374
 003374 005737 001300
 003400 001406
 003402 100002
 003404 000137 003074
 003410 012737 177777 001300
 003416 012737 003432 001106
 003424 012737 003432 001110
 003432
 003432 112737 000001 001102
 003440 012706 001100
 003444 012737 000001 001176
 21
 22 003452 012760 000040 000010
 23
 24
 25
 33 003460 113760 001224 000010
 003466 013737 001224 001240
 003474 005760 000012
 003500 005037 001250
 003504 016037 000010 001126
 003512 012737 000010 001122
 003520 060037 001122
 003524 005037 001124
 003530 013737 001126 001164
 003536 042737 167777 001164
 003544 023737 001124 001164
 003552 001414
 003554 013737 001126 001174
 003562 042737 010000 001174
 003570 053737 001174 001124
 003576 104037
 003600 005137 001250

	003604	000240		64\$:	NOP	
	003606	005737	001250		TST CKERR	; WAS 'NED' SET ?
	003612	001403			.+10	; BR IF NOT
	003614	012760	000040	000010	MOV #CLR, RMCS2(R0)	; ISSUE MASSBUS INIT TO CLEAR 'NED'
	003622	113760	001226	000010	MOV PORTB, RMCS2(R0)	; SELECT PORT B
	003630	013737	001226	001240	MOV PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
	003636	005760	000012		TST RMD5(R0)	; SEE IF DRIVE (PORT B) PRESENT
	003642	005037	001250		CLR CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
	003646	016037	000010	001126	MOV RMCS2(R0), \$BDDAT	; GET CONTENTS OF RMCS2
	003654	012737	000010	001122	MOV #RMCS2, \$BDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
	003662	060037	001122		ADD RO, \$BDADR	; ADD RH/RM BASE ADDRESS
	003666	005037	001124		CLR \$GDDAT	; WHAT REGISTER SHOULD BE
	003672	013737	001126	001164	MOV \$BDDAT, \$TMP0	; MOVE REGISTER CONTENTS TO '\$TMP0'
	003700	042737	167777	001164	BIC #^CNED, \$TMP0	; SAVE SPECIFIED BITS
	003706	023737	001124	001164	CMP \$GDDAT, \$TMP0	; COMPARE THE BITS
	003714	001414			BEQ 66\$; BR IF OK
	003716	013737	001126	001174	MOV \$BDDAT, \$TMP4	; COPY 'BAD DATA'
	003724	042737	010000	001174	BIC #NED, \$TMP4	; CLEAR THE MASKED BITS
	003732	053737	001174	001124	BIS \$TMP4, \$GDDAT	; 'OR' WITH GOOD DATA FOR TYPEOUT
	003740	104037			EMT 37	
	003742	005137	001250		COM CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
	003746	000240		66\$:	NOP	
	003750	005737	001250		TST CKERR	; WAS 'NED' SET ?
	003754	001403			.+10	; BR IF NOT
	003756	012760	000040	000010	MOV #CLR, RMCS2(R0)	; ISSUE MASSBUS INIT TO CLEAR 'NED'
						; CONFIRM THAT DRIVE IS AN RM05, RM03 OR RM02 AND IS DUAL PORTED
34						
35						
36						
40	003764	113760	001224	000010	MOV PORTA, RMCS2(R0)	; SELECT PORT A
	003772	013737	001224	001240	MOV PORTA, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
	004000	005037	001250		CLR CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
	004004	016037	000026	001126	MOV RMDT(R0), \$BDDAT	; GET CONTENTS OF RMDT
	004012	012737	000026	001122	MOV #RMDT, \$BDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
	004020	060037	001122		ADD RO, \$BDADR	; ADD RH/RM BASE ADDRESS
	004024	012737	024027	001124	MOV #024027, \$GDDAT	; WHAT REGISTER SHOULD BE
	004032	022737	024024	001126	CMP #024024, \$BDDAT	; DUAL PORT RM03 ?
	004040	001413			BEQ 68\$; YES !!
	004042	022737	024025	001126	CMP #024025, \$BDDAT	; DUAL PORT RM02 ?
	004050	001407			BEQ 68\$; YES !!
	004052	023737	001124	001126	CMP \$GDDAT, \$BDDAT	; IS THE REGISTER OK ?
	004060	001403			BEQ 68\$; BR IF OK
	004062	104001			EMT 1	
	004064	005137	001250		COM CKERR	; SET THE REGISTER COMPARE ERROR INDICATOR
	004070	000240		68\$:	NOP	
	004072	113760	001226	000010	MOV PORTB, RMCS2(R0)	; SELECT PORT B
	004100	013737	001226	001240	MOV PORTB, PTNBR	; MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
	004106	005037	001250		CLR CKERR	; CLEAR THE 'CHECK ERROR' INDICATOR
	004112	016037	000026	001126	MOV RMDT(R0), \$BDDAT	; GET CONTENTS OF RMDT
	004120	012737	000026	001122	MOV #RMDT, \$BDADR	; FORM REGISTER ADDRESS OF ERROR MESSAGE
	004126	060037	001122		ADD RO, \$BDADR	; ADD RH/RM BASE ADDRESS
	004132	012737	024027	001124	MOV #024027, \$GDDAT	; WHAT REGISTER SHOULD BE
	004140	022737	024024	001126	CMP #024024, \$BDDAT	; DUAL PORT RM03 ?
	004146	001413			BEQ 70\$; YES !!
	004150	022737	024025	001126	CMP #024025, \$BDDAT	; DUAL PORT RM02 ?
	004156	001407			BEQ 70\$; YES !!
	004160	023737	001124	001126	CMP \$GDDAT, \$BDDAT	; IS THE REGISTER OK ?
	004166	001403			BEQ 70\$; BR IF OK

	004170	104001		EMT	1	
	004172	005137	001250	COM	CKERR	;SET THE REGISTER COMPARE ERROR INDICATOR
	004176	000240		70\$: NOP		
41						
42						;VERIFY THROUGH BOTH PORTS THAT THE DRIVE IS ON LINE AND IN NEUTRAL
43						
48	004200	113760	001224	000010	MOVB	PORTA, RMCS2(RO) ;SELECT PORT A
	004206	013737	001224	001240	MOV	PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
	004214	005037	001250		CLR	CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
	004220	016037	000012	001126	MOV	RMDS(RO), \$BDDAT ;GET CONTENTS OF RMDS
	004226	012737	000012	001122	MOV	#RMDS, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
	004234	060037	001122		ADD	RO, \$BDADR ;ADD RH/RM BASE ADDRESS
	004240	012737	001000	001124	MOV	#PGM, \$GDDAT ;WHAT REGISTER SHOULD BE
	004246	013737	001126	001164	MOV	\$BDDAT, \$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
	004254	042737	176777	001164	BIC	#^CPGM, \$TMP0 ;SAVE SPECIFIED BITS
	004262	023737	001124	001164	CMP	\$GDDAT, \$TMP0 ;COMPARE THE BITS
	004270	001414			BEQ	72\$;BR IF OK
	004272	013737	001126	001174	MOV	\$BDDAT, \$TMP4 ;COPY 'BAD DATA'
	004300	042737	001000	001174	BIC	#PGM, \$TMP4 ;CLEAR THE MASKED BITS
	004306	053737	001174	001124	BIS	\$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
	004314	104045			EMT	45
	004316	005137	001250		COM	CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
	004322	000240			72\$: NOP	
	004324	005037	001250		CLR	CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
	004330	016037	000012	001126	MOV	RMDS(RO), \$BDDAT ;GET CONTENTS OF RMDS
	004336	012737	000012	001122	MOV	#RMDS, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
	004344	060037	001122		ADD	RO, \$BDADR ;ADD RH/RM BASE ADDRESS
	004350	012737	010600	001124	MOV	#MOL!DPR!DRY, \$GDDAT ;WHAT REGISTER SHOULD BE
	004356	013737	001126	001164	MOV	\$BDDAT, \$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
	004364	042737	167177	001164	BIC	#^C10600, \$TMP0 ;SAVE SPECIFIED BITS
	004372	023737	001124	001164	CMP	\$GDDAT, \$TMP0 ;COMPARE THE BITS
	004400	001414			BEQ	74\$;BR IF OK
	004402	013737	001126	001174	MOV	\$BDDAT, \$TMP4 ;COPY 'BAD DATA'
	004410	042737	010600	001174	BIC	#10600, \$TMP4 ;CLEAR THE MASKED BITS
	004416	053737	001174	001124	BIS	\$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
	004424	104002			EMT	2
	004426	005137	001250		COM	CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
	004432	000240			74\$: NOP	
	004434	113760	001226	000010	MOVB	PORTB, RMCS2(RO) ;SELECT PORT B
	004442	013737	001226	001240	MOV	PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
	004450	005037	001250		CLR	CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
	004454	016037	000012	001126	MOV	RMDS(RO), \$BDDAT ;GET CONTENTS OF RMDS
	004462	012737	000012	001122	MOV	#RMDS, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
	004470	060037	001122		ADD	RO, \$BDADR ;ADD RH/RM BASE ADDRESS
	004474	012737	001000	001124	MOV	#PGM, \$GDDAT ;WHAT REGISTER SHOULD BE
	004502	013737	001126	001164	MOV	\$BDDAT, \$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
	004510	042737	176777	001164	BIC	#^CPGM, \$TMP0 ;SAVE SPECIFIED BITS
	004516	023737	001124	001164	CMP	\$GDDAT, \$TMP0 ;COMPARE THE BITS
	004524	001414			BEQ	76\$;BR IF OK
	004526	013737	001126	001174	MOV	\$BDDAT, \$TMP4 ;COPY 'BAD DATA'
	004534	042737	001000	001174	BIC	#PGM, \$TMP4 ;CLEAR THE MASKED BITS
	004542	053737	001174	001124	BIS	\$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
	004550	104045			EMT	45
	004552	005137	001250		COM	CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
	004556	000240			76\$: NOP	
	004560	005037	001250		CLR	CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
	004564	016037	000012	001126	MOV	RMDS(RO), \$BDDAT ;GET CONTENTS OF RMDS

```

004572 012737 000012 001122      MOV    #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
004600 060037 001122                ADD    R0,$BDADR   ;ADD RH/RM BASE ADDRESS
004604 012737 010600 001124      MOV    #MOL!DPR!DRY,$GDDAT ;WHAT REGISTER SHOULD BE
004612 013737 001126 001164      MOV    $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
004620 042737 167177 001164      BIC    #^C10600,$TMP0 ;SAVE SPECIFIED BITS
004626 023737 001124 001164      CMP    $GDDAT,$TMP0   ;COMPARE THE BITS
004634 001414                BEQ    78$           ;BR IF OK
004636 013737 001126 001174      MOV    $BDDAT,$TMP4   ;COPY 'BAD DATA'
004644 042737 010600 001174      BIC    #10600,$TMP4   ;CLEAR THE MASKED BITS
004652 053737 001174 001124      BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
004660 104002                EMT    2
004662 005137 001250                COM
004666 000240                CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
78$:  NOP

```

49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
81
82

;VERIFY THAT DRIVE SERIAL NUMBER SEEN THROUGH BOTH PORTS IS THE SAME

```

52 004670 113760 001224 000010      MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
53 004676 016037 000030 001124      MOV    RMSN(R0),$GDDAT ;STORE THE PORT A SERIAL NUMBER
54 004704 113760 001226 000010      MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
55 004712 016037 000030 001126      MOV    RMSN(R0),$BDDAT ;STORE THE PORT B SERIAL NUMBER
56 004720 023737 001124 001126      CMP    $GDDAT,$BDDAT  ;ARE THEY THE SAME ?
57 004726 001406                BEQ    1$           ;BR IF THEY ARE
58 004730 104003                EMT    3
59 004732 032777 100000 174200      BIT    #SW15,@SWR     ;HALT ON ERROR ?
60 004740 001001                BNE    1$           ;BR IF SET - PROGRAM HAS ALREADY HALTED
61 004742 000000                HALT
62 004744 000004                1$:  SCOPE          ;HALT, POSSIBLE CABLE CONNECTION PROBLEM
;LOOP ?

```

```

*****
*TFST 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.
*****

```

```

004746 005737 001300                TST2:  TST    KYBCTL   ;PERFORMING ONLY SINGLE TEST
004746 001406                BEQ    2$           ;BR IF NOT
004754 100002                BPL    1$           ;BR IF JUST ENTERED TEST
004756 000137 003074                JMP    EXEC        ;RETURN & GET NEXT TEST NUMBER
004762 012737 177777 001300      1$:  MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
004770 012737 005004 001106      2$:  MOV    #TEST2,$LPADR ;SETUP SCOPE LOOP ADDRESS
004776 012737 005004 001110      MOV    #TEST2,$LPERR ;SETUP ERROR LOOP ADDRESS
005004                TEST2:  MOVB   #2,$STSTM   ;MOVE #2 TO TEST NUMBER
005004 112737 000002 001102

```



```

005012 012706 001100      MOV    #STACK,SP      ;LOAD THE STACK POINTER
005016 012737 000002 001176  MOV    #2.,$TIMES     ;;DO 2. ITERATIONS

83
151 005024 012737 000240 177776  MOV    #<5*32.>,@#PS  ;SET PRIORITY TO 5 IN CASE LOOPING
005032 005037 001262      CLR    TIMEA          ;CLEAR TIMEOUT VALUE FOR PORT A
005036 005037 001264      CLR    TIMEAP         ;CLEAR UPPER TIMEOUT TOLERANCE
005042 005037 001266      CLR    TIMEAM        ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

005046 005037 001256      CLR    TIME           ;CLEAR THE ELAPSED TIME COUNTER
005052 012737 003720 001260  MOV    #2000.,WATCH  ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT A

005060 113760 001224 000010      MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
005066 013737 001224 001242      MOV    PORTA,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
005074 005060 000006      CLR    RMDA(R0)      ;WRITE RMDA
005100 113760 001226 000010      MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
005106 013737 001226 001240      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005114 013737 001226 001244      MOV    PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
005122 016037 000012 001126      MOV    RMD5(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
005130 010037 001122      MOV    R0,$BDADR     ;RH/RM BASE ADDRESS
005134 062737 000012 001122      ADD    #RMD5,$BDADR  ;GENERATE BAD REGISTER ADDRESS
005142 005037 001124      CLR    $GDDAT        ;REGISTER SHOULD BE ZERO
005146 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER ZERO
005154 001403      BEQ    64$          ;BR IF IT IS
005156 104004      EMT    4
005160 000137 006312      JMP    5$           ;BYPASS REST OF THE SUBTEST
005164
64$:
005164 113760 001224 000010      MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
005172 013737 001224 001240      MOV    PORTA,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005200 016037 000012 001126      MOV    RMD5(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
005206 042737 020001 001126      BIC    #OM!PIP,$BDDAT ;CLEAR DONT CARE BITS
005214 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;EXPECTED STATUS
005222 013737 001124 001166      MOV    $GDDAT,$TMP1  ;USE GOOD DATA AS A MASK
005230 005137 001166      COM    $TMP1         ;COMPLEMENT THE EXPECTED STATUS
005234 013737 001126 001164      MOV    $BDDAT,$TMP0  ;SAVE THE ACTUAL STATUS
005242 043737 001166 001164      BIC    $TMP1,$TMP0   ;CLEAR UNWANTED BITS
005250 023737 001124 001164      CMP    $GDDAT,$TMP0  ;ARE THE EXPECTED STATUS BITS SET ?
005256 001401      BEQ    65$          ;BR IF THEY ARE
005260 104005      EMT    5
005262 000240      NOP

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

005264 113760 001226 000010      MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
005272 013737 001226 001240      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
005300 016046 000046      MOV    RMEC2(R0),-(SP) ;STORE REGISTER RMEC2, PORT B, FOR CHECK
005304 016046 000044      MOV    RMEC1(R0),-(SP) ;STORE REGISTER RMEC1, PORT B, FOR CHECK
005310 016046 000030      MOV    RMSN(R0),-(SP) ;STORE REGISTER RMSN, PORT B, FOR CHECK
005314 016046 000034      MOV    RMDC(R0),-(SP) ;STORE REGISTER RMDC, PORT B, FOR CHECK
005320 016046 000032      MOV    RMOF(R0),-(SP) ;STORE REGISTER RMOF, PORT B, FOR CHECK
005324 016046 000042      MOV    RMER2(R0),-(SP) ;STORE REGISTER RMER2, PORT B, FOR CHECK
005330 016046 000020      MOV    RMLA(R0),-(SP) ;STORE REGISTER RMLA, PORT B, FOR CHECK
005334 016046 000026      MOV    RMDT(R0),-(SP) ;STORE REGISTER RMDT, PORT B, FOR CHECK
005340 016046 000006      MOV    RMDA(R0),-(SP) ;STORE REGISTER RMDA, PORT B, FOR CHECK
  
```

005344 016046 000024 MOV RMMR1(RO),-(SP) ;STORE REGISTER RMMR1, PORT B, FOR CHECK
 005350 016046 000014 MOV RMER1(RO),-(SP) ;STORE REGISTER RMER1, PORT B, FOR CHECK

;WAIT FOR PORT A TO TIMEOUT

005354 005760 000012 1\$: TST RMDS(RO) ;WAIT FOR THE DRIVE TO TIMEOUT
 005360 001006 BNE 2\$;BR WHEN TIMEOUT OCCURS
 005362 005737 001260 TST WATCH ;CHECK WATCH
 005366 001372 BNE 1\$;BR IF NOT ZERO
 005370 104036 EMT 36
 005372 000137 005776 JMP 4\$;BYPASS TIMEOUT TIME CHECK
 005376 012737 000340 177776 2\$: MOV #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
 005404 013737 001256 001262 MOV TIME,TIMEA ;SAVE THE ELAPSED TIME FOR PORT A
 005412 004537 066324 JSR R5,TOLER ;CALCULATE THE TOLERANCE
 005416 001262 .WORD TIMEA ;TIMEOUT VALUE FOR PORT A
 005420 012637 001264 MOV (SP)+,TIMEAP ;+25% TOLERANCE
 005424 012637 001266 MOV (SP)+,TIMEAM ;-25% TOLERANCE

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

005430 023727 001256 000764 CMP TIME,#500. ;WAS MEASURED TIME AT LEAST 500 MS?
 005436 103001 BHIS 3\$;BR IF IT WAS
 005440 104055 EMT 55

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

005442 012737 000240 177776 3\$: MOV #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

005450 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
 005454 012737 000012 001122 MOV #RMDS,\$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
 005462 060037 001122 ADD RO,\$BDADR ;ADD THE I/O BASE ADDRESS
 005466 012737 011600 001124 MOV #MOL!PGM!DPR!DRY,\$GDDAT ;COMPARISON CONSTANT
 005474 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
 005502 016037 000012 001170 MOV RMDS(RO), \$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 005510 042737 024001 001170 BIC #PIP!WRL!OM,\$TMP2 ;CLEAR DONT CARES
 005516 013737 001170 001164 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
 005524 042737 100100 001164 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 005532 113760 001225 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
 005540 016037 000012 001172 MOV RMDS(RO), \$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 005546 042737 024001 001172 BIC #PIP!WRL!OM,\$TMP3 ;CLEAR DONT CARES
 005554 013737 001172 001166 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
 005562 042737 100100 001166 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 005570 023737 001164 001166 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 005576 001006 BNE 66\$;BR IF NOT
 005600 005737 001164 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 005604 001037 BNE 68\$;BR IF NOT
 005606 104046 EMT 46
 005610 000137 005774 JMP 70\$;BYPASS THE REST OF THE CHECKS
 005614 013737 001170 001126 66\$: MOV \$TMP2,\$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
 005622 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 005630 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
 005636 005737 001164 TST \$TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
 005642 001414 BEQ 67\$;BR IF ZERO
 005644 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
 005652 013737 001172 001126 MOV \$TMP3,\$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT

```

005660 113760 001224 000010      MOVB  PORTA, RMCS2(R0)  ;SELECT PORT A.
005666 005737 001166              TST   $TMP1             ;SEE IF STATUS EQ ZERO FROM PORT B.
005672 001004              BNE   68$              ;BR IF NOT
005674 012737 177777 001254 67$:  MOV   #-1, RELERR      ;SET 'RELEASE ERROR' INDICATOR
005702 104022              EMT   22
005704 013737 001170 001126 68$:  MOV   $TMP2, $BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDs READ
005712 013737 001224 001240      MOV   PORTA, PTNBR     ;CHANGE PORT NUMBER
005720 042737 100100 001126      BIC   #ATA.VV, $BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
005726 023737 001124 001126      CMP   $GDDAT, $BDDAT  ;ALL BITS OK ?
005734 001401              BEQ   69$              ;BR IF OK FROM PORT A.
005736 104007              EMT   7
005740 013737 001172 001126 69$:  MOV   $TMP3, $BDDAT    ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
005746 013737 001226 001240      MOV   PORTB, PTNBR    ;CHANGE PORT NUMBER
005754 042737 100100 001126      BIC   #ATA.VV, $BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
005762 023737 001124 001126      CMP   $GDDAT, $BDDAT  ;SEE IF READ OK FROM PORT B.
005770 001401              BEQ   70$              ;BR IF OK
005772 104007              EMT   7
005774 000240 70$:  NOP

```

```

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

```

```

005776 013737 001226 001240 4$:  MOV   PORTB, PTNBR    ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
006004 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMER1
006010 062737 000014 001122      ADD   #RMER1, $BDADR  ;ADDRESS OF RMER1 FOR TYPEOUT
006016 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMER1
006022 001401              BEQ   .+4              ;CONTENTS ZERO ?
006024 104006              EMT   6
006026 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMMR1
006032 062737 000024 001122      ADD   #RMMR1, $BDADR  ;ADDRESS OF RMMR1 FOR TYPEOUT
006040 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMMR1
006044 001401              BEQ   .+4              ;CONTENTS ZERO ?
006046 104006              EMT   6
006050 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMDA
006054 062737 000006 001122      ADD   #RMDA, $BDADR   ;ADDRESS OF RMDA FOR TYPEOUT
006062 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMDA
006066 001401              BEQ   .+4              ;CONTENTS ZERO ?
006070 104006              EMT   6
006072 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMDT
006076 062737 000026 001122      ADD   #RMDT, $BDADR   ;ADDRESS OF RMDT FOR TYPEOUT
006104 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMDT
006110 001401              BEQ   .+4              ;CONTENTS ZERO ?
006112 104006              EMT   6
006114 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMLA
006120 062737 000020 001122      ADD   #RMLA, $BDADR   ;ADDRESS OF RMLA FOR TYPEOUT
006126 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMLA
006132 001401              BEQ   .+4              ;CONTENTS ZERO ?
006134 104006              EMT   6
006136 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMER2
006142 062737 000042 001122      ADD   #RMER2, $BDADR  ;ADDRESS OF RMER2 FOR TYPEOUT
006150 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMER2
006154 001401              BEQ   .+4              ;CONTENTS ZERO ?
006156 104006              EMT   6
006160 010037 001122              MOV   R0, $BDADR      ;BASE ADDRESS FOR REGISTER RMOF
006164 062737 000032 001122      ADD   #RMOF, $BDADR   ;ADDRESS OF RMOF FOR TYPEOUT
006172 012637 001126              MOV   (SP)+, $BDDAT   ;CHECK THE STORED CONTENTS OF RMOF
006176 001401              BEQ   .+4              ;CONTENTS ZERO ?

```

```

006200 104006          EMT      6
006202 010037 001122  MOV     RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMDC
006206 062737 000034 001122  ADD     #RMDC,$BDADR   ;ADDRESS OF RMDC FOR TYPEOUT
006214 012637 001126  MOV     (SP)+,$BDDAT   ;CHECK THE STORED CONTENTS OF RMDC
006220 001401          BEQ     .+4             ;CONTENTS ZERO ?
006222 104006          EMT      6
006224 010037 001122  MOV     RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMSN
006230 062737 000030 001122  ADD     #RMSN,$BDADR   ;ADDRESS OF RMSN FOR TYPEOUT
006236 012637 001126  MOV     (SP)+,$BDDAT   ;CHECK THE STORED CONTENTS OF RMSN
006242 001401          BEQ     .+4             ;CONTENTS ZERO ?
006244 104006          EMT      6
006246 010037 001122  MOV     RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC1
006252 062737 000044 001122  ADD     #RMEC1,$BDADR  ;ADDRESS OF RMEC1 FOR TYPEOUT
006260 012637 001126  MOV     (SP)+,$BDDAT   ;CHECK THE STORED CONTENTS OF RMEC1
006264 001401          BEQ     .+4             ;CONTENTS ZERO ?
006266 104006          EMT      6
006270 010037 001122  MOV     RO,$BDADR      ;BASE ADDRESS FOR REGISTER RMEC2
006274 062737 000046 001122  ADD     #RMEC2,$BDADR  ;ADDRESS OF RMEC2 FOR TYPEOUT
006302 012637 001126  MOV     (SP)+,$BDDAT   ;CHECK THE STORED CONTENTS OF RMEC2
006306 001401          BEQ     .+4             ;CONTENTS ZERO ?
006310 104006          EMT      6
006312 000004          EMT      6
5$:      SCOPE          ;LOOP ?
    
```

152
170
171

```

:*****
:*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.
:*****
    
```

```

006314 005737 001300  TST3:  TST     KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
006314 001406          BEQ     2$           ;BR IF NOT
006322 100002          BPL     1$           ;BR IF JUST ENTERED TEST
006324 000137 003074  JMP     EXEC         ;RETURN & GET NEXT TEST NUMBER
006330 012737 177777 001300 1$:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
006336 012737 006352 001106 2$:  MOV     #TEST3,$LPADR ;SETUP SCOPE LOOP ADDRESS
006344 012737 006352 001110  MOV     #TEST3,$LPERR ;SETUP ERROR LOOP ADDRESS
006352          TEST3:  MOV     #3,$TSTNM    ;MOVE #3 TO TEST NUMBER
006352 112737 000003 001102  MOV     #STACK,SP    ;LOAD THE STACK POINTER
006360 012706 001100          MOV     #2,$TIMES    ;DO 2. ITERATIONS
006364 012737 000002 001176 172  MOV     #<5*32.>,@#PS ;SET PRIORITY TO 5 IN CASE LOOPING
173 006372 012737 000240 177776  CLR     TIMEB        ;CLEAR TIMEOUT VALUE FOR PORT B
006400 005037 001270
    
```

172
173

006404 005037 001272 CLR TIMEBP ;CLEAR UPPER TIMEOUT TOLERANCE
 006410 005037 001274 CLR TIMEBM ;CLEAR LOWER TIMEOUT TOLERANCE

;START THE TIMER

006414 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
 006420 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000. MS

;SEIZE THE DRIVE THROUGH PORT B

006426 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 006434 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
 006442 005060 000006 CLR RMDA(R0) ;WRITE RMDA
 006446 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
 006454 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006462 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
 006470 016037 000012 001126 MOV RMD5(R0), \$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
 006476 010037 001122 MOV R0, \$BDADR ;RH/RM BASE ADDRESS
 006502 062737 000012 001122 ADD #RMD5, \$BDADR ;GENERATE BAD REGISTER ADDRESS
 006510 005037 001124 CLR \$GDDAT ;REGISTER SHOULD BE ZERO
 006514 023737 001124 001126 CMP \$GDDAT, \$BDDAT ;IS THE REGISTER ZERO
 006522 001403 BEQ 64\$;BR IF IT IS
 006524 104004 EMT 4
 006526 000137 007660 JMP 5\$;BYPASS REST OF THE SUBTEST
 006532 64\$:
 006532 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 006540 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006546 016037 000012 001126 MOV RMD5(R0), \$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
 006554 042737 020001 001126 BIC #OM!PIP, \$BDDAT ;CLEAR DONT CARE BITS
 006562 012737 011600 001124 MOV #MOL!PGM.DPR!DRY, \$GDDAT ;EXPECTED STATUS
 006570 013737 001124 001166 MOV \$GDDAT, \$TMP1 ;USE GOOD DATA AS A MASK
 006576 005137 001166 COM \$TMP1 ;COMPLEMENT THE EXPECTED STATUS
 006602 013737 001126 001164 MOV \$BDDAT, \$TMP0 ;SAVE THE ACTUAL STATUS
 006610 043737 001166 001164 BIC \$TMP1, \$TMP0 ;CLEAR UNWANTED BITS
 006616 023737 001124 001164 CMP \$GDDAT, \$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
 006624 001401 BEQ 65\$;BR IF THEY ARE
 006626 104005 EMT 5
 006630 65\$:
 006630 000240 NOP

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

006632 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
 006640 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 006646 016046 000046 MOV RMEC2(R0), -(SP) ;STORE REGISTER RMEC2, PORT A, FOR CHECK
 006652 016046 000044 MOV RMEC1(R0), -(SP) ;STORE REGISTER RMEC1, PORT A, FOR CHECK
 006656 016046 000030 MOV RMSN(R0), -(SP) ;STORE REGISTER RMSN, PORT A, FOR CHECK
 006662 016046 000034 MOV RMDC(R0), -(SP) ;STORE REGISTER RMDC, PORT A, FOR CHECK
 006666 016046 000032 MOV RMOF(R0), -(SP) ;STORE REGISTER RMOF, PORT A, FOR CHECK
 006672 016046 000042 MOV RMER2(R0), -(SP) ;STORE REGISTER RMER2, PORT A, FOR CHECK
 006676 016046 000020 MOV RMLA(R0), -(SP) ;STORE REGISTER RM A, PORT A, FOR CHECK
 006702 016046 000026 MOV RMDT(R0), -(SP) ;STORE REGISTER RMDT, PORT A, FOR CHECK
 006706 016046 000006 MOV RMDA(R0), -(SP) ;STORE REGISTER RMDA, PORT A, FOR CHECK
 006712 016046 000024 MOV RMMR1(R0), -(SP) ;STORE REGISTER RMMR1, PORT A, FOR CHECK
 006716 016046 000014 MOV RMER1(R0), -(SP) ;STORE REGISTER RMER1, PORT A, FOR CHECK

;WAIT FOR PORT B TO TIMEOUT

```

006722 005760 000012      1$:   TST   RMDS(R0)      ;WAIT FOR THE DRIVE TO TIMEOUT
006726 001006              BNE   2$           ;BR WHEN TIMEOUT OCCURS
006730 005737 001260      TST   WATCH        ;CHECK WATCH
006734 001372              BNE   1$           ;BR IF NOT ZERO
006736 104036              FMT   36
006740 000137 007344      JMP   4$           ;BYPASS TIMEOUT TIME CHECK
006744 012737 000340 177776 2$:   MOV   #<7*32.>,@#PS ;SET PRIORITY TO 7 TO STOP CLOCK
006752 013737 001256 001270  MOV   TIME,TIMEB   ;SAVE THE ELAPSED TIME FOR PORT B
006760 004537 066324      JSR   RS,TOLER     ;CALCULATE THE TOLERANCE
006764 001270              .WORD TIMEB        ;TIMEOUT VALUE FOR PORT B
006766 012637 001272      MOV   (SP)+,TIMEBP ;+25% TOLERANCE
006772 012637 001274      MOV   (SP)+,TIMEBM ;-25% TOLERANCE

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

006776 023727 001256 000764  CMP   TIME,#500.   ;WAS MEASURED TIME AT LEAST 500 MS?
007004 103001              BHS   3$           ;BR IF IT WAS
007006 104055              EMT   55

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

007010 012737 000240 177776 3$:   MOV   #<5*32.>,@#PS ;RESTORE PRIORITY TO 5

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

007016 005037 001254              CLR   RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
007022 012737 000012 001122  MOV   #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
007030 060037 001122      ADD   R0,$BDADR    ;ADD THE I/O BASE ADDRESS
007034 012737 011600 001124  MOV   #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
007042 113760 001224 000010  MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
007050 016037 000012 001170  MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
007056 042737 024001 001170  BIC   #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
007064 013737 001170 001164  MOV   $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
007072 042737 100100 001164  BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007100 113760 001226 000010  MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
007106 016037 000012 001172  MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
007114 042737 024001 001172  BIC   #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
007122 013737 001172 001166  MOV   $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
007130 042737 100100 001166  BIC   #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
007136 023737 001164 001166  CMP   $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
007144 001006              BNE   66$         ;BR IF NOT
007146 005737 001164      TST   $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
007152 001037              BNE   68$         ;BR IF NOT
007154 104046              EMT   46
007156 000137 007342      JMP   70$         ;BYPASS THE REST OF THE CHECKS
007162 013737 001170 001126 66$:  MOV   $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
007170 013737 001226 001240  MOV   PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007176 113760 001226 000010  MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
007204 005737 001164      TST   $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
007210 001414              BEQ   67$         ;BR IF ZERO
007212 013737 001224 001240  MOV   PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
007220 013737 001172 001126  MOV   $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
007226 113760 001224 000010  MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
007234 005737 001166      TST   $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
007240 001004              BNE   68$         ;BR IF NOT
007242 012737 177777 001254 67$:  MOV   #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
007250 104022              EMT   22
    
```

```

007252 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
007260 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
007266 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007274 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
007302 001401 BEQ 69$ ;BR IF OK FROM PORT A.
007304 104007 EMT 7
007306 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
007314 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
007322 042737 100100 001126 BIC #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
007330 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
007336 001401 BEQ 70$ ;BR IF OK
007340 104007 EMT 7
007342 000240 70$: NOP
  
```

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

```

007344 013737 001224 001240 4$: MOV PORTA,PTNBR ;CHANGE 'PORT NUMBER' TO THE OPPOSITE PORT
007352 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER1
007356 062737 000014 001122 ADD #RMER1,$BDADR ;ADDRESS OF RMER1 FOR TYPEOUT
007364 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER1
007370 001401 BEQ .+4 ;CONTENTS ZERO ?
007372 104006 EMT 6
007374 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMMR1
007400 062737 000024 001122 ADD #RMMR1,$BDADR ;ADDRESS OF RMMR1 FOR TYPEOUT
007406 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMMR1
007412 001401 BEQ .+4 ;CONTENTS ZERO ?
007414 104006 EMT 6
007416 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDA
007422 062737 000006 001122 ADD #RMDA,$BDADR ;ADDRESS OF RMDA FOR TYPEOUT
007430 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDA
007434 001401 BEQ .+4 ;CONTENTS ZERO ?
007436 104006 EMT 6
007440 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDT
007444 062737 000026 001122 ADD #RMDT,$BDADR ;ADDRESS OF RMDT FOR TYPEOUT
007452 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDT
007456 001401 BEQ .+4 ;CONTENTS ZERO ?
007460 104006 EMT 6
007462 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMLA
007466 062737 000020 001122 ADD #RMLA,$BDADR ;ADDRESS OF RMLA FOR TYPEOUT
007474 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMLA
007500 001401 BEQ .+4 ;CONTENTS ZERO ?
007502 104006 EMT 6
007504 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMER2
007510 062737 000042 001122 ADD #RMER2,$BDADR ;ADDRESS OF RMER2 FOR TYPEOUT
007516 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMER2
007522 001401 BEQ .+4 ;CONTENTS ZERO ?
007524 104006 EMT 6
007526 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMOF
007532 062737 000032 001122 ADD #RMOF,$BDADR ;ADDRESS OF RMOF FOR TYPEOUT
007540 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMOF
007544 001401 BEQ .+4 ;CONTENTS ZERO ?
007546 104006 EMT 6
007550 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMDC
007554 062737 000034 001122 ADD #RMDC,$BDADR ;ADDRESS OF RMDC FOR TYPEOUT
007562 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMDC
007566 001401 BEQ .+4 ;CONTENTS ZERO ?
  
```



```

007570 104006 EMT 6
007572 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMSN
007576 062737 000030 001122 ADD #RMSN,$BDADR ;ADDRESS OF RMSN FOR TYPEOUT
007604 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMSN
007610 001401 BEQ .+4 ;CONTENTS ZERO ?
007612 104006 EMT 6
007614 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC1
007620 062737 000044 001122 ADD #RMEC1,$BDADR ;ADDRESS OF RMEC1 FOR TYPEOUT
007626 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC1
007632 001401 BEQ .+4 ;CONTENTS ZERO ?
007634 104006 EMT 6
007636 010037 001122 MOV RO,$BDADR ;BASE ADDRESS FOR REGISTER RMEC2
007642 062737 000046 001122 ADD #RMEC2,$BDADR ;ADDRESS OF RMEC2 FOR TYPEOUT
007650 012637 001126 MOV (SP)+,$BDDAT ;CHECK THE STORED CONTENTS OF RMEC2
007654 001401 BEQ .+4 ;CONTENTS ZERO ?
007656 104006 EMT 6
007660 000004 5$: SCOPE ;LOOP ?
    
```

174
188
189

```

:*****
:*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.
:*****
    
```

```

007662 005737 001300 TST4: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
007666 001406 BEQ 2$ ;BR IF NOT
007670 100002 BPL 1$ ;BR IF JUST ENTERED TEST
007672 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
007676 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
007704 012737 007720 001106 2$: MOV #TEST4,$LPADR ;SETUP SCOPE LOOP ADDRESS
007712 012737 007720 001110 MOV #TEST4,$LPERR ;SETUP ERROR LOOP ADDRESS
007720 112737 000004 001102 TEST4: MOVB #4,$TSTNM ;MOVE #4 TO TEST NUMBER
007726 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
007732 012737 000012 001176 MOV #10,$TIMES ;DO 10. ITERATIONS
    
```

190
223

```

;START THE TIMER
007740 005037 001256 CLR TIME ;CLEAR THE ELAPSED TIME COUNTER
007744 012737 003720 001260 MOV #2000.,WATCH ;SET WATCH TO 2000. MS
;SEIZE THE DRIVE AND SET VOLUME VALID
;SEIZE THE DRIVE THROUGH PORT A
007752 113760 001224 000010 MOVB PORTA,RMCS2(RO) ;SELECT PORT A
    
```



```

007760 013737 001224 001242      MOV   PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
007766 005060 000012                CLR   RMDS(R0)      ;WRITE RMDS
007772 013737 001226 001244      MOV   PORTB,OPPRT  ;'OPPOSITE' PORT ADDRESS
010000 012760 000021 000000      MOV   #21,RMCS1(R0) ;SET VOLUME VALID
010006 005037 001250                CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
010012 016037 000012 001126      MOV   RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
010020 012737 000012 001122      MOV   #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010026 060037 001122                ADD   R0,$BDADR    ;ADD RH/RM BASE ADDRESS
010032 012737 000100 001124      MOV   #VV,$GDDAT   ;WHAT REGISTER SHOULD BE
010040 013737 001126 001164      MOV   $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010046 042737 177677 001164      BIC   #^CVV,$TMP0  ;SAVE SPECIFIED BITS
010054 023737 001124 001164      CMP   $GDDAT,$TMP0 ;COMPARE THE BITS
010062 001414                BEQ   66$          ;BR IF OK
010064 013737 001126 001174      MOV   $BDDAT,$TMP4 ;COPY 'BAD DATA'
010072 042737 000100 001174      BIC   #VV,$TMP4    ;CLEAR THE MASKED BITS
010100 053737 001174 001124      BIS   $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010106 104013                EMT   13
010110 005137 001250                COM   CKERR
010114 000240                NOP
010116 012760 000040 000010      MOV   #CLR,RMCS2(R0) ;CLEAR DRIVE
                                     ;RELEASE THE DRIVE FROM PORT A

010124 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
010132 013737 001224 001240      MOV   PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010140 012760 000013 000000      MOV   #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
                                     ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

010146 005037 001254                CLR   RELERR       ;CLEAR THE 'RELEASE ERROR' INDICATOR
010152 012737 000012 001122      MOV   #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
010160 060037 001122                ADD   R0,$BDADR    ;ADD THE I/O BASE ADDRESS
010164 012737 011600 001124      MOV   #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
010172 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A.
010200 016037 000012 001170      MOV   RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
010206 042737 024001 001170      BIC   #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
010214 013737 001170 001164      MOV   $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
010222 042737 100100 001164      BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010230 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B.
010236 016037 000012 001172      MOV   RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
010244 042737 024001 001172      BIC   #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
010252 013737 001172 001166      MOV   $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
010260 042737 100100 001166      BIC   #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
010266 023737 001164 001166      CMP   $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
010274 001006                BNE   68$         ;BR IF NOT
010276 005737 001164                TST   $TMP0
010302 001037                BNE   70$         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
010304 104046                EMT   46
010306 000137 010472                JMP   72$         ;BYPASS THE REST OF THE CHECKS
010312 013737 001170 001126      MOV   $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
010320 013737 001226 001240      MOV   PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010326 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B.
010334 005737 001164                TST   $TMP0
010340 001414                BEQ   69$         ;SEE IF STATUS EQ 0 FROM PORT A.
010342 013737 001224 001240      MOV   PORTA,PTNBR  ;BR IF ZERO
010350 013737 001172 001126      MOV   $TMP3,$BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
010356 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;'BAD DATA' FOR ERROR TYPE OUT
                                     ;SELECT PORT A.

```

```

010364 005737 001166          TST      $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
010370 001004          BNE      70$          ;BR IF NOT
010372 012737 177777 001254 69$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
010400 104022          EMT      22
010402 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
010410 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
010416 042737 100100 001126      BIC     #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
010424 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
010432 001401          BEQ     71$          ;BR IF OK FROM PORT A.
010434 104007          EMT      7
010436 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
010444 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
010452 042737 100100 001126      BIC     #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
010460 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
010466 001401          BEQ     72$          ;BR IF OK
010470 104007          EMT      7
010472 000240          NOP
010474 005737 001254          TST     RELERR       ;DID DRIVE RETURN TO NEUTRAL ?
010500 001402          BEQ     .+6         ;BR IF IN NEUTRAL
010502 000137 010756          JMP     1$          ;GO WAIT FOR DRIVE TO TIMEOUT
010506 113760 001224 000010      MOV     PORTA, RMCS2(R0) ;SELECT PORT A
010514 013737 001224 001240      MOV     PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010522 005037 001250          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
010526 016037 000012 001126      MOV     RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
010534 012737 000012 001122      MOV     #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010542 060037 001122          ADD     R0,$BDADR   ;ADD RH/RM BASE ADDRESS
010546 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
010552 013737 001126 001164      MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010560 042737 077777 001164      BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
010566 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
010574 001414          BEQ     73$          ;BR IF OK
010576 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
010604 042737 100000 001174      BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
010612 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010620 104017          EMT      17
010622 005137 001250          COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
010626 000240          NOP
010630 113760 001226 000010      MOV     PORTB, RMCS2(R0) ;SELECT PORT B
010636 013737 001226 001240      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
010644 005037 001250          CLR     CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
010650 016037 000012 001126      MOV     RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
010656 012737 000012 001122      MOV     #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
010664 060037 001122          ADD     R0,$BDADR   ;ADD RH/RM BASE ADDRESS
010670 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
010674 013737 001126 001164      MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
010702 042737 077777 001164      BIC     #^CATA,$TMP0 ;SAVE SPECIFIED BITS
010710 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
010716 001414          BEQ     75$          ;BR IF OK
010720 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
010726 042737 100000 001174      BIC     #ATA,$TMP4  ;CLEAR THE MASKED BITS
010734 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
010742 104017          EMT      17
010744 005137 001250          COM     CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
010750 000240          NOP
010752 000137 011010          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

010756
 010756 113760 001226 000010
 010764 013737 001226 001240
 010772 005760 000012
 010776 001004
 011000 005737 001260
 011004 001364
 011006 104036
 011010 000004

```

1$:      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
        MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
        TST   RMDS(R0) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
        BNE   2$ ;BR WHEN DRIVE RELEASED
        TST   WATCH ;CHECK THE WATCH
        BNE   1$ ;BR IF NOT ZERO
        EMT   36
2$:      SCOPE ;LOOP ?
    
```

224
 238
 239

```

:*****
:*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.
:*****
    
```

011012
 011012 005737 001300
 011016 001406
 011020 100002
 011022 000137 003074
 011026 012737 177777 001300
 011034 012737 011050 001106
 011042 012737 011050 001110
 011050
 011050 112737 000005 001102
 011056 012706 001100
 011062 012737 000012 001176

```

TST5:
        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   #TEST5, $LPADR ;SETUP SCOPE LOOP ADDRESS
        MOV   #TEST5, $LPERR ;SETUP ERROR LOOP ADDRESS
TEST5:
        MOVB  #5, $STSTM ;MOVE #5 TO TEST NUMBER
        MOV   #STACK, SP ;LOAD THE STACK POINTER
        MOV   #10., $TIMES ;DO 10. ITERATIONS
    
```

240
 241

;START THE TIMER

011070 005037 001256
 011074 012737 003720 001260

```

        CLR   TIME ;CLEAR THE ELAPSED TIME COUNTER
        MOV   #2000., WATCH ;SET WATCH TO 2000. MS
    
```

;SEIZE THE DRIVE AND SET VOLUME VALID

;SEIZE THE DRIVE THROUGH PORT B

011102 113760 001226 000010
 011110 013737 001226 001242
 011116 005060 000012
 011122 013737 001224 001244
 011130 012760 000021 000000
 011136 005037 001250
 011142 016037 000012 001126

```

        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
        MOV   #21, RMCS1(R0) ;SET VOLUME VALID
        CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
        MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
    
```

```

011150 012737 000012 001122      MOV      #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011156 060037 001122                ADD      R0,$BDADR ;ADD RM/RM BASE ADDRESS
011162 012737 000100 001124      MOV      #VV,$GDDAT ;WHAT REGISTER SHOULD BE
011170 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011176 042737 177677 001164      BIC      #^CVV,$TMP0 ;SAVE SPECIFIED BITS
011204 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
011212 001414                BEQ      66$ ;BR IF OK
011214 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
011222 042737 000100 001174      BIC      #VV,$TMP4 ;CLEAR THE MASKED BITS
011230 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR 'YPEOUT
011236 104013                EMT      13
011240 005137 001250                COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
011244 000240                NOP
011246 012760 000040 000010      MOV      #CLR,RMCS2(R0) ;CLEAR DRIVE

;RELEASE THE DRIVE FROM PORT B

011254 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
011262 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011270 012760 000013 000000      MOV      #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

011276 005037 001254                CLR      RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
011302 012737 000012 001122      MOV      #RMD5,$BDADR ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
011310 060037 001122                ADD      R0,$BDADR ;ADD THE I/O BASE ADDRESS
011314 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
011322 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
011330 016037 000012 001170      MOV      RMD5(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
011336 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
011344 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
011352 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011360 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
011366 016037 000012 001172      MOV      RMD5(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
011374 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
011402 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
011410 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
011416 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
011424 001006                BNE      68$ ;BR IF NOT
011426 005737 001164                TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
011432 001037                BNE      70$ ;BR IF NOT
011434 104046                EMT      46
011436 000137 011622                JMP      72$ ;BYPASS THE REST OF THE CHECKS
011442 013737 001170 001126      MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
011450 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011456 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
011464 005737 001164                TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
011470 001414                BEQ      69$ ;BR IF ZERO
011472 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
011500 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
011506 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
011514 005737 001166                TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
011520 001004                BNE      70$ ;BR IF NOT
011522 012737 177777 001254      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
011530 104022                EMT      22
011532 013737 001170 001126      MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
011540 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
  
```

```

011546 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011554 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
011562 001401 BEQ 71$ ;BR IF OK FROM PORT A.
011564 104007 EMT 7
011566 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
011574 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
011602 042737 100100 001126 BIC #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
011610 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
011616 001401 BEQ 72$ ;BR IF OK
011620 104007 EMT 7
011622 000240 72$: NOP
011624 C05737 001254 TST RELERR ;DID DRIVE RETURN TO NEUTRAL ?
011630 001402 BEQ .+6 ;BR IF IN NEUTRAL
011632 000137 012106 JMP 1$ ;GO WAIT FOR DRIVE TO TIMEOUT
011636 113760 001226 000010 MOVB PCRTB, RMCS2(R0) ;SELECT PORT B
011644 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011652 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
011656 016037 000012 001126 MOV RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
011664 012737 000012 001122 MOV #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
011672 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
011676 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
011702 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
011710 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
011716 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
011724 001414 BEQ 73$ ;BR IF OK
011726 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
011734 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
011742 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
011750 104017 EMT 17
011752 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
011756 000240 73$: NOP
011760 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
011766 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
011774 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
012000 016037 000012 001126 MOV RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
012006 012737 000012 001122 MOV #RMD5,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
012014 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
012020 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
012024 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
012032 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
012040 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
012046 001414 BEQ 75$ ;BR IF OK
012050 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
012056 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
012064 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
012072 104017 EMT 17
012074 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
012100 000240 75$: NOP
012102 000137 012140 JMP 2$ ;GO CHECK FOR LOOP ON ERROR

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

012106 113760 001224 000010 1$: MOVB PORTA, RMCS2(R0) ;SELECT PORT A
012114 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
012122 005760 000012 TST RMD5(R0) ;WAIT FOR TIMEOUT TO RELEASE DRIVE
  
```

012126 001004
 012130 005737 C01260
 012134 001364
 012136 104036
 012140 000004

BNE 2\$;BR WHEN DRIVE RELEASED
 TST WATCH ;CHECK THE WATCH
 BNE 1\$;BR IF NOT ZERO
 EMT 36
 2\$: SCOPE ;LOOP ?

246
 255
 256

 *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.
 *

012142
 012142 005737 001300
 012146 001406
 012150 100002
 012152 000137 003074
 012156 012737 177777 001300
 012164 012737 012200 001106
 012172 012737 012200 001110
 012200
 012200 112737 000006 001102
 012206 012706 001100
 012212 012737 000012 001176
 257
 268 012220 113760 001224 000010
 012226 013737 001224 001240
 012234 013737 001224 001242

TST6:
 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 #TEST6,\$LPADR ;SETUP SCOPE LOOP ADDRESS
 MOV #TEST6,\$LPERR ;SETUP ERROR LOOP ADDRESS
 TEST6:
 MOVB #6,\$STNM ;MOVE #6 TO TEST NUMBER
 MOV #STACK,\$SP ;LOAD THE STACK POINTER
 MOV #10,\$TIMES ;DO 10. ITERATIONS
 MOVB PORTA,\$RMCS2(R0) ;SELECT PORT A
 MOV PORTA,\$PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 MOV PORTA,\$SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE

012242 012760 000013 000000

;ISSUE A RELEASE COMMAND
 MOV #13,\$RMCS1(R0) ;ISSUE A RELEASE COMMAND

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

012250 005037 001254
 012254 012737 000012 001122
 012262 060037 001122
 012266 012737 011700 001124
 012274 113760 001224 000010
 012302 016037 000012 001170
 012310 042737 024001 001170
 012316 013737 001170 001164
 012324 042737 100100 001164
 012332 113760 001226 000010
 012340 016037 000012 001172
 012346 042737 024001 001172
 012354 013737 001172 001166
 012362 042737 100100 001166
 012370 023737 001164 001166
 012376 001006
 012400 005737 001164
 012404 001045
 012406 104046

CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
 MOV #RMDS,\$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
 ADD R0,\$BDADR ;ADD THE I/O BASE ADDRESS
 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;COMPARISON CONSTANT
 MOVB PORTA,\$RMCS2(R0) ;SELECT PORT A.
 MOV RMDS(R0),\$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
 BIC #PIP!WRL!OM,\$TMP2 ;CLEAR DONT CARES
 MOV \$TMP2,\$TMP0 ;COPY IT INTO '\$TMP0'
 BIC #ATA!VV,\$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 MOVB PORTB,\$RMCS2(R0) ;SELECT PORT B.
 MOV RMDS(R0),\$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
 BIC #PIP!WRL!OM,\$TMP3 ;CLEAR DONT CARES
 MOV \$TMP3,\$TMP1 ;COPY IT INTO '\$TMP1'
 BIC #ATA!VV,\$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
 CMP \$TMP0,\$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
 BNE 64\$;BR IF NOT
 TST \$TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
 BNE 66\$;BR IF NOT
 EMT 46

```

012410 000137 012610          JMP      68$          ;BYPASS THE REST OF THE CHECKS
012414 013737 001170 001126 64$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
012422 013737 001226 001240      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012430 113760 001226 000010      MOV     PORTB,RMCS2(R0) ;SELECT PORT B.
012436 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
012442 001414          BEQ     65$          ;BR IF ZERO
012444 013737 001224 001240      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
012452 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
012460 113760 001224 000010      MOV     PORTA,RMCS2(R0) ;SELECT PORT A.
012466 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
012472 001012          BNE     66$          ;BR IF NOT
012474 012737 177777 001254 65$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
012502 012760 000011 000000      MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
012510 012760 000013 000000      MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
012516 104030          EMT     30
012520 013737 001170 001126 66$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
012526 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
012534 042737 100000 001126      BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
012542 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
012550 001401          BEQ     67$          ;BR IF OK FROM PORT A.
012552 104007          EMT     7
012554 013737 001172 001126 67$:  MOV     $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
012562 013737 001226 001240      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
012570 042737 100000 001126      BIC     #ATA,$BDDAT  ;DON'T CHECK THE ATTN BIT
012576 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
012604 001401          BEQ     68$          ;BR IF OK
012606 104007          EMT     7
012610 000240          68$:  NOP
012612 000004          SCOPE          ;LOOP ?

```

269
278
279

```

*****
*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.
*
*****

```

```

012614 005737 001300          TST7:  TST     KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
012614 001406          BEQ     2$           ;BR IF NOT
012620 001406          BPL     1$           ;BR IF JUST ENTERED TEST
012622 100002          JMP     EXEC         ;RETURN & GET NEXT TEST NUMBER
012624 000137 003074          1$:  MOV     #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
012630 012737 177777 001300 2$:  MOV     #TEST7,$LPADR ;SETUP SCOPE LOOP ADDRESS
012636 012737 012652 001106      MOV     #TEST7,$LPERR ;SETUP ERROR LOOP ADDRESS
012644 012737 012652 001110          TEST7:  MOV     #7,$STNM     ;MOVE #7 TO TEST NUMBER
012652 112737 000007 001102      MOV     #STACK,SP   ;LOAD THE STACK POINTER
012660 012706 001100          MOV     #10,$TIMES  ;;DO 10. ITERATIONS
012664 012737 000012 001176          MOV     PORTB,RMCS2(R0) ;SELECT PORT B
280 012672 113760 001226 000010      MOV     PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
281 012700 013737 001226 001240      MOV     PORTB,SEIZPT ;ADDR OF PORT WHICH WILL ISSUE RELEASE
012706 013737 001226 001242

```

280
281


```

012714 012760 C00013 000000 ;ISSUE A RELEASE COMMAND
                                MOV #13,RMCS1(R0) ;ISSUE A RELEASE COMMAND

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

012722 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
012726 012737 000012 001122 MOV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
012734 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
012740 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
012746 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
012754 016037 000012 001170 MOV RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
012762 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
012770 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
012776 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013004 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
013012 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
013020 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
013026 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
013034 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
013042 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
013050 001006 BNE 64$ ;BR IF NOT
013052 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
013056 001045 BNE 66$ ;BR IF NOT
013060 104046 EMT 46
013062 000137 013262 JMP 68$
013066 013737 001170 001126 64$: MOV $TMP2,$BDAT ;BYPASS THE REST OF THE CHECKS
013074 013737 001226 001240 MOV PORTB,PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
013102 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013110 005737 001164 TST $TMP0 ;SELECT PORT B.
013114 001414 BEQ 65$ ;SEE IF STATUS EQ 0 FROM PORT A.
013116 013737 001224 001240 MOV PORTA,PTNBR ;BR IF ZERO
013124 013737 001172 001126 MOV $TMP3,$BDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
013132 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;'BAD DATA' FOR ERROR TYPE OUT
013140 005737 001166 TST $TMP1 ;SELECT PORT A.
013144 001012 BNE 66$ ;SEE IF STATUS EQ ZERO FROM PORT B.
013146 012737 177777 001254 65$: MOV #-1,RELERR ;BR IF NOT
013154 012760 000011 000000 MOV #11,RMCS1(R0) ;SET 'RELEASE ERROR' INDICATOR
013162 012760 000013 000000 MOV #13,RMCS1(R0) ;CLEAR THE DRIVE
013170 104030 EMT 30 ;RELEASE THE DRIVE
013172 013737 001170 001126 66$: MOV $TMP2,$BDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
013200 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
013206 042737 100000 001126 BIC #ATA,$BDAT ;DON'T CHECK THE ATTN BIT
013214 023737 001124 001126 CMP $GDDAT,$BDAT ;ALL BITS OK ?
013222 001401 BEQ 67$ ;BR IF OK FROM PORT A.
013224 104007 EMT 7
013226 013737 001172 001126 67$: MOV $TMP3,$BDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
013234 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
013242 042737 100000 001126 BIC #ATA,$BDAT ;DON'T CHECK THE ATTN BIT
013250 023737 001124 001126 CMP $GDDAT,$BDAT ;SEE IF READ OK FROM PORT B.
013256 001401 BEQ 68$ ;BR IF OK
013260 104007 EMT 7
013262 000240 68$: NOP
013264 000004 SCOPE ;LOOP ?

```

282
301
302

*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 RMD5.
- *****
- 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.
- *****

013266
 013266 005737 001300
 013272 001406
 013274 100002
 013276 000137 003074
 013302 012737 177777 001300
 013310 012737 013324 001106
 013316 012737 013324 001110
 013324
 013324 112737 000010 001102
 013332 012706 001100
 013336 012737 000012 001176

303
 327

```

TST10:
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      #TEST10,$LPADR ;SETUP SCOPE LOOP ADDRESS
          MOV      #TEST10,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST10:
          MOVB     #10,$STSTM   ;MOVE #10 TO TEST NUMBER
          MOV      #STACK,$SP   ;LOAD THE STACK POINTER
          MOV      #10,$TIMES   ;;DO 10. ITERATIONS
    
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

013344 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
013352 005060 000012              CLR      RMD5(R0)       ;SEIZE THE DRIVE
013356 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
013364 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
013372 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
013400 005060 000012              CLR      RMD5(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
013404 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
013412 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
    
```

;SEIZE THE DRIVE THROUGH PORT B

```

013420 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
013426 013737 001226 001242      MOV      PORTB,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
013434 005060 000012              CLR      RMD5(R0)       ;WRITE RMD5
013440 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
013446 013737 001224 001240      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013454 013737 001224 001244      MOV      PORTA,OPPR    ;'OPPOSITE' PORT ADDRESS
013462 016037 000012 001126      MOV      RMD5(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
013470 010037 001122              MOV      R0,$BDADR      ;RH/RM BASE ADDRESS
013474 062737 000012 001122      ADD      #RMD5,$BDADR   ;GENERATE BAD REGISTER ADDRESS
013502 005037 001124              CLP      $GDDAT         ;REGISTER SHOULD BE ZERO
013506 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;IS THE REGISTER ZERO
013514 001403              BEQ      64$            ;BR IF IT IS
013516 104004              EMT      4
    
```

```

013520 000137 014530          JMP      1$          ;BYPASS REST OF THE SUBTEST
013524          64$:    MOV      PORTB, RMCS2(R0) ;SELECT PORT B
013524 113760 001226 000010    MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013532 013737 001226 001240    MOV      RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
013540 016037 000012 001126    BIC      #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
013546 042737 020001 001126    MOV      #MOL.PGM.DPR.DRY!VV, $GDDAT ;EXPECTED STATUS
013554 012737 011700 001124    MOV      $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
013562 013737 001124 001166    COM      $TMP1 ;COMPLEMENT THE EXPECTED STATUS
013570 005137 001166          MOV      $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
013574 013737 001126 001164    BIC      $TMP1, $TMP0 ;CLEAR UNWANTED BITS
013602 043737 001166 001164    CMP      $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
013610 023737 001124 001164    BEQ      65$          ;BR IF THEY ARE
013616 001401          EMT      5
013620 104005          65$:    NOP
013622 000240

```

;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT A

```

013624 113760 001224 000010    MOV      PORTA, RMCS2(R0) ;SELECT PORT A
013632 013737 001224 001240    MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
013640 012760 000013 000000    MOV      #13, RMCS1(R0) ;ISSUE A RELEASE COMMAND THROUGH PORT A

```

;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT B

```

013646 005037 001250          CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
013652 016037 000012 001126    MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
013660 012737 000012 001122    MOV      #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
013666 060037 001122          ADD      R0, $BDADR ;ADD RH/RM BASE ADDRESS
013672 005037 001124          CLR      $GDDAT ;WHAT REGISTER SHOULD BE
013676 023737 001124 001126    CMP      $GDDAT, $BDDAT ;IS THE REGISTER OK ?
013704 001403          BEQ      66$          ;BR IF OK
013706 104010          EMT      10
013710 005137 001250          COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
013714 016037 000000 001126    66$:    MOV      RMCS1(R0), $BDDAT ;GET THE CONTENTS OF RHCS1
013722 012737 000000 001122    MOV      #RMCS1, $BDADR ;FORM ADDRESS OF REGISTER
013730 060037 001122          ADD      R0, $BDADR ;ADDRESS BASE
013734 032737 020000 001126    BIT      #MCPE, $BDDAT ;IS 'MCPE' SET ?
013742 001404          BEQ      67$          ;BR IF NOT
013744 104011          EMT      11
013746 012760 040000 000000    67$:    MOV      #TRE, RMCS1(R0) ;CLEAR 'MCPE'
013754 000240          NOP
013756 005737 001250          TST      CKERR ;WAS RMDS NON ZERO ?
013762 001402          BEQ      +6 ;CONTENTS OF RMDS SEEN BY PORT A
013764 000137 014530          JMP      1$          ;DRIVE IN NEUTRAL, BYPASS REST OF TEST

```

;RELEASE THE DRIVE FROM PORT B

```

013770 113760 001226 000010    MOV      PORTB, RMCS2(R0) ;SELECT PORT B
013776 013737 001226 001240    MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014004 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

```

014012 005037 001254          CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
014016 012737 111700 001124    MOV      #ATA!MOL!PGM.DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
014024 012737 000012 001122    MOV      #RMDS, $BDADR ;REGISTER ADDRESS INCREMENT
014032 060037 001122          ADD      R0, $BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT

```

```

014036 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
014044 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014052 016037 000012 001164      MOV   RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT A
014060 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
014066 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014074 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT B
014102 001404                BEQ   68$ ;BR IF STATUS FROM PORT B ZERO
014104 005737 001164                TST  $TMP0 ;IS STATUS FROM PORT A ZERO ?
014110 001401                BEQ   68$ ;BR IF ZERO
014112 104031                EMT  31
014114 013737 001164 001126 68$: MOV  $TMP0, $BDDAT ;CHECK STATUS FROM PORT A
014122 013737 001224 001240      MOV   PORTA, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
014130 023737 001124 001126      CMP  $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
014136 001401                BEQ   69$ ;BR IF OK
014140 104027                EMT  27
014142 000240                69$: NOP
  
```

;RELEASE THE DRIVE FROM PORT A

```

014144 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
014152 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014160 012760 000013 000000      MOV   #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

014166 005037 001254                CLR  RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
014172 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
014200 060037 001122                ADD  R0, $BDADR ;ADD THE I/O BASE ADDRESS
014204 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
014212 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
014220 016037 000012 001170      MOV  RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
014226 042737 024001 001170      BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
014234 013737 001170 001164      MOV  $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
014242 042737 100100 001164      BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014250 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
014256 016037 000012 001172      MOV  RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
014264 042737 024001 001172      BIC  #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
014272 013737 001172 001166      MOV  $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
014300 042737 100100 001166      BIC  #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
014306 023737 001164 001166      CMP  $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
014314 001006                BNE  70$ ;BR IF NOT
014316 005737 001164                TST  $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
014322 001045                BNE  72$ ;BR IF NOT
014324 104046                EMT  46
014326 000137 014526                JMP  74$
014332 013737 001170 001126 70$: MOV  $TMP2, $BDDAT ;BYPASS THE REST OF THE CHECKS
014340 013737 001226 001240      MOV  PORTB, PTNBR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
014346 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014354 005737 001164                TST  $TMP0 ;SELECT PORT B.
014360 001414                BEQ  71$ ;SEE IF STATUS EQ 0 FROM PORT A.
014362 013737 001224 001240      MOV  PORTA, PTNBR ;BR IF ZERO
014370 013737 001172 001126      MOV  $TMP3, $BDDAT ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
014376 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;'BAD DATA' FOR ERROR TYPE OUT
014404 005737 001166                TST  $TMP1 ;SELECT PORT A.
014410 001012                BNE  72$ ;SEE IF STATUS EQ ZERO FROM PORT B.
014412 012737 177777 001254 71$: MOV  #-1, RELERR ;BR IF NOT
014420 012760 000011 000000      MOV  #11, RMCS1(R0) ;SET 'RELEASE ERROR' INDICATOR
                                ;CLEAR THE DRIVE
  
```

```

014426 012760 000013 000000      MOV    #13,RMCS1(R0) ;RELEASE THE DRIVE
014434 104026      EMT    26
014436 013737 001170 001126 72$:  MOV    $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
014444 013737 001224 001240      MOV    PORTA,PTNBR ;CHANGE PORT NUMBER
014452 042737 100000 001126      BIC    #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
014460 023737 001124 001126      CMP    $GDDAT,$BDDAT ;ALL BITS OK ?
014466 001401      BEQ    73$ ;BR IF OK FROM PORT A.
014470 104007      EMT    7
014472 013737 001172 001126 73$:  MOV    $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
014500 013737 001226 001240      MOV    PORTB,PTNBR ;CHANGE PORT NUMBER
014506 042737 100000 001126      BIC    #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
014514 023737 001124 001126      CMP    $GDDAT,$BDDAT ;SEF IF READ OK FROM PORT B.
014522 001401      BEQ    74$ ;BR IF OK
014524 104007      EMT    7
014526 000240      74$:  NOP
014530 000004      1$:  SCOPE ;LOOP ?
  
```

328
347
348

```

*****
*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.
*****
  
```

```

014532 005737 001300      TST    KYBCTL ;PERFORMING ONLY SINGLE TEST ?
014532 001406      BEQ    2$ ;BR IF NOT
014540 100002      BPL    1$ ;BR IF JUST ENTERED TEST
014542 000137 003074      JMP    EXEC ;RETURN & GET NEXT TEST NUMBER
014546 012737 177777 001300 1$:  MOV    #-1,KYBCTL ;SET SINGLE TEST INDICATOR
014554 012737 014570 001106 2$:  MOV    #TEST11,$LPADR ;SETUP SCOPE LOOP ADDRESS
014562 012737 014570 001110      MOV    #TEST11,$LPERR ;SETUP ERROR LOOP ADDRESS
014570
014570 112737 000011 001102 TEST11: MOVB   #11,$TSTNM ;MOVE #11 TO TEST NUMBER
014576 012706 001100      MOV    #STACK,SP ;LOAD THE STACK POINTER
014602 012737 000012 001176      MOV    #10,$TIMES ;DO 10. ITERATIONS
  
```

349
350

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

014610 113760 001224 000010      MOVB   PORTA,RMCS2(R0) ;SELECT PORT #A
014616 005060 000012      CLR    RMDS(R0) ;SEIZE THE DRIVE
014622 012760 000011 000000      MOV    #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
014630 012760 000013 000000      MOV    #13,RMCS1(R0) ;RELEASE THE DRIVE
  
```

```

014636 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT #B
014644 005060 000012              CLR   RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
014650 012760 000011 000000      MOV   #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
014656 012760 000013 000000      MOV   #13,RMCS1(R0)  ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT A

014664 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
014672 013737 001224 001242      MOV   PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
014700 005060 000012              CLR   RMDS(R0)       ;WRITE RMDS
014704 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
014712 013737 001226 001240      MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
014720 013737 001226 001244      MOV   PORTB,OPPRT    ;'OPPOSITE' PORT ADDRESS
014726 016037 000012 001126      MOV   RMDS(R0),SBDDAT ;SEE IF DRIVE SEIZED BY PORT A
014734 010037 001122              MOV   R0,$BDADR      ;RH/RM BASE ADDRESS
014740 062737 000012 001122      ADD   #RMDS,$BDADR   ;GENERATE BAD REGISTER ADDRESS
014746 005037 001124              CLR   $GDDAT         ;REGISTER SHOULD BE ZERO
014752 023737 001124 001126      CMP   $GDDAT,$BDAT   ;IS THE REGISTER ZERO
014760 001403              BEQ   64$           ;BR IF IT IS
014762 104004              EMT   4
014764 000137 015774              JMP   1$           ;BYPASS REST OF THE SUBTEST
014770

64$:
014770 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
014776 013737 001224 001240      MOV   PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015004 016037 000012 001126      MOV   RMDS(R0),SBDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
015012 042737 020001 001126      BIC   #OM!PIP,$BDAT  ;CLEAR DONT CARE BITS
015020 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
015026 013737 001124 001166      MOV   $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
015034 005137 001166              COM   $TMP1          ;COMPLEMENT THE EXPECTED STATUS
015040 013737 001126 001164      MOV   $BDAT,$TMP0    ;SAVE THE ACTUAL STATUS
015046 043737 001166 001164      BIC   $TMP1,$TMP0    ;CLEAR UNWANTED BITS
015054 023737 001124 001164      CMP   $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
015062 001401              BEQ   65$           ;BR IF THEY ARE
015064 104005              EMT   5
015066 000240              NOP

                                ;TRY TO EXECUTE A RELEASE COMMAND THROUGH PORT B

015070 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
015076 013737 001226 001240      MOV   PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015104 012760 000013 000000      MOV   #13,RMCS1(R0)  ;ISSUE A RELEASE COMMAND THROUGH PORT B

                                ;VERIFY THAT THE DRIVE IS STILL SEIZED BY PORT A

015112 005037 001250              CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
015116 016037 000012 001126      MOV   RMDS(R0),SBDDAT ;GET CONTENTS OF RMDS
015124 012737 000012 001122      MOV   #RMDS,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
015132 060037 001122              ADD   R0,$BDADR      ;ADD RH/RM BASE ADDRESS
015136 005037 001124              CLR   $GDDAT         ;WHAT REGISTER SHOULD BE
015142 023737 001124 001126      CMP   $GDDAT,$BDAT   ;IS THE REGISTER OK ?
015150 001403              BEQ   66$           ;BR IF OK
015152 104010              EMT   10
015154 005137 001250              COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
015160 016037 000000 001126      MOV   RMCS1(R0),SBDDAT ;GET THE CONTENTS OF RMCS1
015166 012737 000000 001122      MOV   #RMCS1,$BDADR  ;FORM ADDRESS OF REGISTER
015174 060037 001122              ADD   R0,$BDADR      ;ADDRESS BASE
015200 032737 020000 001126      BIT   #MCPE,$BDAT    ;IS 'MCPE' SET ?

```

```

015206 001404          BEQ    67$          ;BR IF NOT
015210 104011          EMT    11
015212 012760 040000 000000      MOV    #TRE, RMCS1(R0) ;CLEAR 'MCPE'
015220 000240          NOP
015222 005737 001250      TST    CKERR          ;WAS RMDS NON ZERO ?
015226 001402          BEQ    .+6           ;CONTENTS OF RMDS SEEN BY PORT B
015230 000137 015774      JMP    1$           ;DRIVE IN NEUTRAL, BYPASS REST OF TEST
  
```

;RELEASE THE DRIVE FROM PORT A

```

015234 113760 001224 000010      MOV    PORTA, RMCS2(R0) ;SELECT PORT A
015242 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015250 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

```

015256 005037 001254          CLR    RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
015262 012737 111700 001124      MOV    #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
015270 012737 000012 001122      MOV    #RMDS,$BDADR ;REGISTER ADDRESS INCREMENT
015276 060037 001122          ADD    R0,$BDADR    ;REGISTER BASE ADDRESS FOR TYPEOUT
015302 113760 001226 000010      MOV    PORTB, RMCS2(R0) ;SELECT PORT B
015310 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015316 016037 000012 001164      MOV    RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT B
015324 113760 001224 000010      MOV    PORTA, RMCS2(R0) ;SELECT PORT A
015332 013737 001224 001240      MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015340 016037 000012 001126      MOV    RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT A
015346 001404          BEQ    68$          ;BR IF STATUS FROM PORT A ZERO
015350 005737 001164          TST    $TMP0        ;IS STATUS FROM PORT B ZERO ?
015354 001401          BEQ    68$          ;BR IF ZERO
015356 104031          EMT    31
015360 013737 001164 001126      MOV    $TMP0,$BDDAT ;CHECK STATUS FROM PORT B
015366 013737 001226 001240      MOV    PORTB, PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
015374 023737 001124 001126      CMP    $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
015402 001401          BEQ    69$          ;BR IF OK
015404 104027          EMT    27
01540C 000240          NOP
69$:
  
```

;RELEASE THE DRIVE FROM PORT B

```

015410 113760 001226 000010      MOV    PORTB, RMCS2(R0) ;SELECT PORT B
015416 013737 001226 001240      MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
015424 012760 000013 000000      MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

015432 005037 001254          CLR    RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
015436 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
015444 060037 001122          ADD    R0,$BDADR    ;ADD THE I/O BASE ADDRESS
015450 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
015456 113760 001224 000010      MOV    PORTA, RMCS2(R0) ;SELECT PORT A.
015464 016037 000012 001170      MOV    RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
015472 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
015500 013737 001170 001164      MOV    $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
015506 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015514 113760 001226 000010      MOV    PORTB, RMCS2(R0) ;SELECT PORT B.
015522 016037 000012 001172      MOV    RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
015530 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
  
```

```

015536 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
015544 042737 100100 001166      BIC      #ATA.VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
015552 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
015560 001006                BNE      70$              ;BR IF NOT
015562 005737 001164                TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
015566 001045                BNE      72$              ;BR IF NOT
015570 104046                EMT      46
015572 000137 015772                JMP      74$              ;BYPASS THE REST OF THE CHECKS
015576 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
015604 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015612 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
015620 005737 001164                TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
015624 001414                BEQ      71$              ;BR IF ZERO
015626 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
015634 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
015642 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
015650 005737 001164                TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
015654 001012                BNE      72$              ;BR IF NOT
015656 012737 177777 001254 71$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
015664 012760 000011 000000      MOV      #11,RMCS1(R0)   ;CLEAR THE DRIVE
015672 012760 000013 000000      MOV      #13,RMCS1(R0)   ;RELEASE THE DRIVE
015700 104026                EMT      26
015702 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
015710 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
015716 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
015724 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
015732 001401                BEQ      73$              ;BR IF OK FROM PORT A.
015734 104007                EMT      7
015736 013737 001172 001126 73$:  MOV      $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
015744 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
015752 042737 100000 001126      BIC      #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
015760 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
015766 001401                BEQ      74$              ;BR IF OK
015770 104007                EMT      7
015772 000240                NOP
015774 000004                1$:  SCOPE                    ;LOOP ?
  
```

351
371
372

```

*****
*TFST 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.
*
*****
  
```


373
407

```

*****
TST12:
015776 005737 001300      TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
015776 001406              BEQ      2$          ;BR IF NOT
016002 001406              BPL      1$          ;BR IF JUST ENTERED TEST
016004 100002              JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
016006 000137 003074      1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
016012 012737 177777 001300 2$:      MOV      #TEST12,$LPADR ;SETUP SCOPE LOOP ADDRESS
016020 012737 016034 001106      MOV      #TEST12,$LPERR ;SETUP ERROR LOOP ADDRESS
016026 012737 016034 001110
016034
016034 112737 000012 001102 TEST12:  MOVB     #12,$TSTNM ;MOVE #12 TO TEST NUMBER
016042 012706 001100      MOV      #STACK,SP ;LOAD THE STACK POINTER
016046 012737 000012 001176      MOV      #10, $TIMES ;DO 10. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS
016054 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
016062 005060 000012              CLR      RMDS(R0)      ;SEIZE THE DRIVE
016066 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
016074 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
016102 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
016110 005060 000012              CLR      RMDS(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
016114 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
016122 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A
016130 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
016136 013737 001224 001242      MOV      PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS
016144 005060 000012              CLR      RMDS(R0)      ;WRITE RMDS
016150 013737 001226 001244      MOV      PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS

;FORCE AN ERROR
016156 012760 177777 000014      MOV      #-1,RMER1(R0) ;SET ERROR BITS
016164 012760 000013 000000      MOV      #13,RMCS1(R0) ;ISSUE A RELEASE COMMAND
016172 005037 001250              CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
016176 016037 000000 001126      MOV      RMCS1(R0), $BDDAT ;GET CONTENTS OF RMCS1
016204 012737 000000 001122      MOV      #RMCS1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016212 060037 001122              ADD      R0,$BDADR    ;ADD RH/RM BASE ADDRESS
016216 012737 004012 001124      MOV      #4012,$GDDAT ;WHAT REGISTER SHOULD BE
016224 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
016232 042737 173765 001164      BIC     #^C4012,$TMP0 ;SAVE SPECIFIED BITS
016240 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
016246 001414              BEQ     66$          ;BR IF OK
016250 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
016256 042737 004012 001174      BIC     #4012,$TMP4 ;CLEAR THE MASKED BITS
016264 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
016272 104025              EMT     25
016274 005137 001250              COM     CKERR
016300 000240              NOP
016302 005737 001250      66$:    TST     CKERR        ;DID 'GO' BIT RESET ?
016306 001002              BNE     +6          ;BR IF NOT
016310 000137 016350              JMP     1$          ;'GO' BIT RESET
016314 012760 000040 000010      MOV     #CLR,RMCS2(R0) ;INIT THE RH/RM
016322 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
  
```



```
016330 013737 001224 001240      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016336 012760 000013 000000      MOV    #13,RMCS1(RO) ;RELEASE THE DRIVE THROUGH PORT A
016344 000137 017114                JMP    2$ ;BYPASS THE REST OF THE TEST
```

;VERIFY THAT DRIVE IS STILL SEIZED BY PORT A

```
016350 113760 001226 000010      1$:   MOVB   PORTB,RMCS2(RO) ;SELECT PORT B
016350 013737 001226 001240      MOV    PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016364 005037 001250                CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
016370 016037 000012 001126      MOV    RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
016376 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016404 060037 001122      ADD    RO,$BDADR ;ADD RH/RM BASE ADDRESS
016410 005037 001124                CLR    $GDDAT ;WHAT REGISTER SHOULD BE
016414 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
016422 001403                BEQ    68$ ;BR IF OK
016424 104024                EMT    24
016426 005137 001250      68$:   COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
016432 000240                NOP
016434 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
016442 013737 001224 001240      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016450 005037 001250                CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
016454 016037 000014 001126      MOV    RMER1(RO), $BDDAT ;GET CONTENTS OF RMER1
016462 012737 000014 001122      MOV    #RMER1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
016470 060037 001122      ADD    RO,$BDADR ;ADD RH/RM BASE ADDRESS
016474 012737 177777 001124      MOV    #177777,$GDDAT ;WHAT REGISTER SHOULD BE
016502 023737 001124 001126      CMP    $GDDAT,$BDDAT ;IS THE REGISTER OK ?
016510 001403                BEQ    70$ ;BR IF OK
016512 104010                EMT    10
016514 005137 001250      70$:   COM    CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
016520 000240                NOP
```

;CLEAR THE ERRORS THROUGH PORT A

```
016522 012760 000011 000000      MOV    #11,RMCS1(RO) ;ISSUE A DRIVE CLEAR
```

;RELEASE THE DRIVE FROM PORT A

```
016530 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A
016536 013737 001224 001240      MOV    PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
016544 012760 000013 000000      MOV    #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
016552 005037 001254                CLR    RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
016556 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
016564 060037 001122      ADD    RO,$BDADR ;ADD THE I/O BASE ADDRESS
016570 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
016576 113760 001224 000010      MOVB   PORTA,RMCS2(RO) ;SELECT PORT A.
016604 016037 000012 001170      MOV    RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
016612 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
016620 013737 001170 001164      MOV    $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
016626 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
016634 113760 001226 000010      MOVB   PORTB,RMCS2(RO) ;SELECT PORT B.
016642 016037 000012 001172      MOV    RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
016650 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
016656 013737 001172 001166      MOV    $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
```

```

016664 042737 100100 001166 BIC #ATA,VV,$TMP1 :CLEAR PORT DEPENDENT BITS FROM THE COPY
016672 023737 001164 001166 CMP $TMP0,$TMP1 :IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
016700 001006 BNE 72$ :BR IF NOT
016702 005737 001164 TST $TMP0 :REGISTERS ARE THE SAME: ARE THEY ZERO ?
016706 001045 BNE 74$ :BR IF NOT
016710 104046 EMT 46
016712 000137 017112 JMP 76$ :BYPASS THE REST OF THE CHECKS
016716 013737 001170 001126 72$: MOV $TMP2,$BDDAT :SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
016724 013737 001226 001240 MOV PORTB,PTNBR :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016732 113760 001226 000010 MOVB PORTB,RMCS2(R0) :SELECT PORT B.
016740 005737 001164 TST $TMP0 :SEE IF STATUS EQ 0 FROM PORT A.
016744 001414 BEQ 73$ :BR IF ZERO
016746 013737 001224 001240 MOV PORTA,PTNBR :SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
016754 013737 001172 001126 MOV $TMP3,$BDDAT :'BAD DATA' FOR ERROR TYPE OUT
016762 113760 001224 000010 MOVB PORTA,RMCS2(R0) :SELECT PORT A.
016770 005737 001166 TST $TMP1 :SEE IF STATUS EQ ZERO FROM PORT B.
016774 001012 BNE 74$ :BR IF NOT
016776 012737 177777 001254 73$: MOV #-1,RELERR :SET 'RELEASE ERROR' INDICATOR
017004 012760 000011 000000 MOV #11,RMCS1(R0) :CLEAR THE DRIVE
C17012 012760 000013 000000 MOV #13,RMCS1(R0) :RELEASE THE DRIVE
017020 104026 EMT 26
017022 013737 001170 001126 74$: MOV $TMP2,$BDDAT :LOOK FOR BIT FAILURES WHEN RMDs READ
017030 013737 001224 001240 MOV PORTA,PTNBR :CHANGE PORT NUMBER
017036 042737 100000 001126 BIC #ATA,$BDDAT :DON'T CHECK THE ATTN BIT
017044 023737 001124 001126 CMP $GDDAT,$BDDAT :ALL BITS OK ?
017052 001401 BEQ 75$ :BR IF OK FROM PORT A.
017054 104007 EMT 7
017056 013737 001172 001126 75$: MOV $TMP3,$BDDAT :CHECK RMDs FOR BIT FAILURES - FROM PORT B.
017064 013737 001226 001240 MOV PORTB,PTNBR :CHANGE PORT NUMBER
017072 042737 100000 001126 BIC #ATA,$BDDAT :DON'T CHECK THE ATTN BIT
017100 023737 001124 001126 CMP $GDDAT,$BDDAT :SEE IF READ OK FROM PORT B.
017106 001401 BEQ 76$ :BR IF OK
017110 104007 EMT 7
017112 000240 76$: NOP
017114 000004 2$: SCOPE :LOOP ?
  
```

408
428
429

```

:*****
:*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.
:*
:*****
  
```

017116
 017116 J05737 001300
 017122 001406
 017124 100002
 017126 000137 003074
 017132 012737 177777 001300
 017140 012737 017154 001106
 017146 012737 017154 001110
 017154
 017154 112737 000013 001102
 017162 012706 001100
 017166 012737 000012 001176

TST13:
 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\$: MO #1,KYBCTL ;SET SINGLE TEST INDICATOR
 2\$: MOV #TEST13,\$LPADR ;SETUP SCOPE LOOP ADDRESS
 MOV #TEST13,\$LPERR ;SETUP ERROR LOOP ADDRESS
 TEST13:
 MOVB #13,\$STSTM ;MOVE #13 TO TEST NUMBER
 MOV #STACK,SP ;LOAD THE STACK POINTER
 MOV #10.,\$TIMES ;DO 10. ITERATIONS

430
 431

;CLEAR ATTENTION BITS FOR BOTH PORTS

017174 113760 001224 000010
 017202 005060 000012
 017206 012760 000011 000000
 017214 012760 000013 000000
 017222 113760 001226 000010
 017230 005060 000012
 017234 012760 000011 000000
 017242 012760 000013 000000

MOVB PORTA, RMCS2(R0) ;SELECT PORT #A
 CLR RMD5(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 RMD5(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

017250 113760 001226 000010
 017256 013737 001226 001242
 017264 005060 000012
 017270 013737 001224 001244

MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
 CLR RMD5(R0) ;WRITE RMD5
 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS

;FORCE AN ERROR

017276 012760 177777 000014
 017304 012760 000013 000000
 017312 005037 001250
 017316 016037 000000 001126
 017324 012737 000000 001122
 017332 060037 001122
 017336 012737 004012 001124
 017344 013737 001126 001164
 017352 042737 173765 001164
 017360 023737 001124 001164
 017366 001414
 017370 013737 001126 001174
 017376 042737 004012 001174
 017404 053737 001174 001124
 017412 104025
 017414 005137 001250
 017420 000240
 017422 005737 001250
 017426 001002
 017430 000137 017470
 017434 012760 000040 000010
 017442 113760 001226 000010
 017450 013737 001226 001240

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
 CMP \$GDDAT, \$TMP0 ;COMPARE THE BITS
 BEQ 66\$;BR IF OK
 MOV \$BDDAT, \$TMP4 ;COPY 'BAD DATA'
 BIC #4012, \$TMP4 ;CLEAR THE MASKED BITS
 BIS \$TMP4, \$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
 EMT 25
 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
 66\$: NOP
 TST CKERR ;DID 'GO' BIT RESET ?
 BNE .+6 ;BR IF NOT
 JMP 1\$;'GO' BIT RESET
 MOV #CLR, RMCS2(R0) ;INIT THE RH/RM
 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

017456 012760 000013 000000
017464 000137 020234

MOV #13, RMCS1(R0) ;RELEASE THE DRIVE THROUGH PORT B
JMP 2\$;BYPASS THE REST OF THE TEST

;VERIFY THAT DRIVE 13 STILL SEIZED BY PORT B

017470
017470 113760 001224 000010
017476 013737 001224 001240
017504 005037 001250
017510 016037 000012 001126
017516 012737 000012 001122
017524 060037 001122
017530 005037 001124
017534 023737 001124 001126
017542 001403
017544 104024
017546 005137 001250
017552 000240
017554 113760 001226 000010
017562 013737 001226 001240
017570 005037 001250
017574 016037 000014 001126
017602 012737 000014 001122
017610 060037 001122
017614 012737 177777 001124
017622 023737 001124 001126
017630 001403
017632 104010
017634 005137 001250
017640 000240

1\$:
MOV#B PORTA, RMCS2(R0) ;SELECT PORT A
MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS(R0), \$BDDAT ;GET CONTENTS OF RMDS
MOV #RMDS, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, \$BDADR ;ADD RH/RM BASE ADDRESS
CLR \$GDDAT ;WHAT REGISTER SHOULD BE
CMP \$GDDAT, \$BDDAT ;IS THE REGISTER OK ?
BEQ 68\$;BR IF OK
EMT 24
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
68\$:
NOP
MOV#B PORTB, RMCS2(R0) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMER1(R0), \$BDDAT ;GET CONTENTS OF RMER1
MOV #RMER1, \$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, \$BDADR ;ADD RH/RM BASE ADDRESS
MOV #177777, \$GDDAT ;WHAT REGISTER SHOULD BE
CMP \$GDDAT, \$BDDAT ;IS THE REGISTER OK ?
BEQ 70\$;BR IF OK
EMT 10
COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
70\$:
NOP

;CLEAR THE ERRORS THROUGH PORT B

017642 012760 000011 000000

MOV #11, RMCS1(R0) ;ISSUE A DRIVE CLEAR

;RELEASE THE DRIVE FROM PORT B

017650 113760 001226 000010
017656 013737 001226 001240
017664 012760 000013 000000

MOV#B PORTB, RMCS2(R0) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

017672 005037 001254
017676 012737 000012 001122
017704 060037 001122
017710 012737 011700 001124
017716 113760 001224 000010
017724 016037 000012 001170
017732 042737 024001 001170
017740 013737 001170 001164
017746 042737 100100 001164
017754 113760 001226 000010
017762 016037 000012 001172
017770 042737 024001 001172
017776 013737 001172 001166
020004 042737 100100 001166

CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
MOV #RMDS, \$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
ADD R0, \$BDADR ;ADD THE I/O BASE ADDRESS
MOV #MOL!PGM!DPR!DRY!VV, \$GDDAT ;COMPARISON CONSTANT
MOV#B PORTA, RMCS2(R0) ;SELECT PORT A.
MOV RMDS(R0), \$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
BIC #PIP!WRL!OM, \$TMP2 ;CLEAR DONT CARES
MOV \$TMP2, \$TMP0 ;COPY IT INTO '\$TMP0'
BIC #ATA!VV, \$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
MOV#B PORTB, RMCS2(R0) ;SELECT PORT B.
MOV RMDS(R0), \$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
BIC #PIP!WRL!OM, \$TMP3 ;CLEAR DONT CARES
MOV \$TMP3, \$TMP1 ;COPY IT INTO '\$TMP1'
BIC #ATA.VV, \$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY

```

020012 023737 001164 001166      CMP      $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
020020 001006                      BNE      72$             ;BR IF NOT
020022 005737 001164              TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
020026 001045                      BNE      74$             ;BR IF NOT
020030 104046                      EMT      46
020032 000137 020232              JMP      76$             ;BYPASS THE REST OF THE CHECKS
020036 013737 001170 001126 72$:  MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
020044 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020052 113760 001226 000010      MOVVB   PORTB,RMCS2(R0)  ;SELECT PORT B.
020060 005737 001164              TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
020064 001414                      BEQ      73$             ;BR IF ZERO
020066 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
020074 013737 001172 001126      MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
020102 113760 001224 000010      MOVVB   PORTA,RMCS2(R0)  ;SELECT PORT A.
020110 005737 001166              TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
020114 001012                      BNE      74$             ;BR IF NOT
020116 012737 177777 001254 73$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
020124 012760 000011 000000      MOV      #11,RMCS1(R0)   ;CLEAR THE DRIVE
020132 012760 000013 000000      MOV      #13,RMCS1(R0)   ;RELEASE THE DRIVE
020140 104026                      EMT      26
020142 013737 001170 001126 74$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
020150 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
020156 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
020164 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
020172 001401                      BEQ      75$             ;BR IF OK FROM PORT A.
020174 104007                      EMT      7
020176 013737 001172 001126 75$:  MOV      $TMP3,$BDDAT     ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
020204 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
020212 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
020220 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
020226 001401                      BEQ      76$             ;BR IF OK
020230 104007                      EMT      7
020232 000240 76$:              NOP
020234 000004 2$:              SCOPE                    ;LOOP ?
  
```

432
451
452

```

*****
*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.
*
*****
  
```

```

020236
020236 005737 001300
  
```

```

TST14:      TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
  
```

```

020242 001406 BEQ 2$ :BR IF NOT
020244 100002 BPL 1$ :BR IF JUST ENTERED TEST
020246 000137 003074 JMP EXEC :RETURN & GET NEXT TEST NUMBER
020252 012737 177777 001300 1$: MOV #-1,KYBCTL :SET SINGLE TEST INDICATOR
020260 012737 020274 001106 2$: MOV #TEST14,$LPADR :SETUP SCOPE LOOP ADDRESS
020266 012737 020274 001110 MOV #TEST14,$LPERR :SETUP ERROR LOOP ADDRESS
020274 TEST14:
020274 112737 000014 001102 MOVB #14,$STSTM :MOVE #14 TO TEST NUMBER
020302 012706 001100 MOV #STACK,SP :LOAD THE STACK POINTER
020306 012737 000012 001176 MOV #10.,$TIMES ;;DO 10. ITERATIONS

```

453
483

;SEIZE THE DRIVE THROUGH PORT A

```

020314 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
020322 013737 001224 001242 MOV PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
020330 005060 000012 CLR RMDS(R0) ;WRITE RMDS
020334 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
020342 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020350 013737 001226 001244 MOV PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
020356 016037 000012 001126 MOV RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
020364 010037 001122 MOV RO, $BDADR ;RH/RM BASE ADDRESS
020370 062737 000012 001122 ADD #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
020376 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
020402 023737 001124 001126 CMP $GDDAT, $BDDAT ;IS THE REGISTER ZERO
020410 001403 BEQ 64$ ;BR IF IT IS
020412 104004 EMT 4
020414 000137 021634 JMP 1$ ;BYPASS REST OF THE SUBTEST
020420 64$:
020420 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
020426 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020434 016037 000012 001126 MOV RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
020442 042737 020001 001126 BIC #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
020450 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
020456 013737 001124 001166 MOV $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
020464 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
020470 013737 001126 001164 MOV $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
020476 043737 001166 001164 BIC $TMP1, $TMP0 ;CLEAR UNWANTED BITS
020504 023737 001124 001164 CMP $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
020512 001401 BEQ 65$ ;BR IF THEY ARE
020514 104005 EMT 5
020516 000240 65$: NOP

```

;DRIVE CLEAR THROUGH PORT A FIRST

```

020520 012760 000011 000000 MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR THROUGH PORT A

```

;VERIFY THAT DRIVE STILL SEIZED BY PORT A

```

020526 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
020534 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020542 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
020546 016037 000012 001126 MOV RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
020554 012737 000012 001122 MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020562 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
020566 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
020572 012737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'

```

```

020600 042737 100000 001164      BIC      #^C77777,$TMP0      ;SAVE SPECIFIED BITS
020606 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
020614 001414                      BEQ      66$                ;BR IF OK
020616 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
020624 042737 077777 001174      BIC      #77777,$TMP4      ;CLEAR THE MASKED BITS
020632 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
020640 104033                      EMT      33
020642 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
020646 000240                      66$:  NOP
020650 113760 001224 000010      MOVB     PORTA, RMCS2(R0)    ;SELECT PORT A
020656 013737 001224 001240      MOV      PC:TA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
020664 005037 001250                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
020670 016037 000012 001126      MOV      RMDS(R0), $BDDAT    ;GET CONTENTS OF RMDS
020676 012737 000012 001122      MOV      #RMDS, $BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
020704 060037 001122                      ADD      RO, $BDADR         ;ADD RH/RM BASE ADDRESS
020710 012737 011700 001124      MOV      #MOL.PGM.DPR!DRY.VV,$GDDAT ;WHAT REGISTER SHOULD BE
020716 013737 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
020724 042737 100000 001164      BIC      #^C77777,$TMP0      ;SAVE SPECIFIED BITS
020732 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
020740 001414                      BEQ      68$                ;BR IF OK
020742 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
020750 042737 077777 001174      BIC      #77777,$TMP4      ;CLEAR THE MASKED BITS
020756 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
020764 104033                      EMT      33
020766 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
020772 000240                      68$:  NOP

```

;NOW ISSUE MASSBUS INIT

```

020774 012760 000040 000010      MOV      #CLR, RMCS2(R0)    ;ISSUE MASSBUS INIT

```

;CONFIRM THAT DRIVE STILL SEIZED BY PORT A

```

021002 113760 001226 000010      MOVB     PORTB, RMCS2(R0)    ;SELECT PORT B
021010 013737 001226 001240      MOV      PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021016 005037 001250                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
021022 016037 000012 001126      MOV      RMDS(R0), $BDDAT    ;GET CONTENTS OF RMDS
021030 012737 000012 001122      MOV      #RMDS, $BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021036 060037 001122                      ADD      RO, $BDADR         ;ADD RH/RM BASE ADDRESS
021042 005037 001124                      CLR      $GDDAT            ;WHAT REGISTER SHOULD BE
021046 013737 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
021054 042737 100000 001164      BIC      #^C77777,$TMP0      ;SAVE SPECIFIED BITS
021062 023737 001124 001164      CMP      $GDDAT,$TMP0      ;COMPARE THE BITS
021070 001414                      BEQ      70$                ;BR IF OK
021072 013737 001126 001174      MOV      $BDDAT,$TMP4      ;COPY 'BAD DATA'
021100 042737 077777 001174      BIC      #77777,$TMP4      ;CLEAR THE MASKED BITS
021106 053737 001174 001124      BIS      $TMP4,$GDDAT      ;'OR' WITH GOOD DATA FOR TYPEOUT
021114 104034                      EMT      34
021116 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
021122 000240                      70$:  NOP
021124 113760 001224 000010      MOVB     PORTA, RMCS2(R0)    ;SELECT PORT A
021132 013737 001224 001240      MOV      PORTA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021140 005037 001250                      CLR      CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
021144 016037 000012 001126      MOV      RMDS(R0), $BDDAT    ;GET CONTENTS OF RMDS
021152 012737 000012 001122      MOV      #RMDS, $BDADR      ;FORM REGISTER ADDRESS OF ERROR MESSAGE
021160 060037 001122                      ADD      RO, $BDADR         ;ADD RH/RM BASE ADDRESS
021164 012737 011700 001124      MOV      #MOL.PGM.DPR!DRY.VV,$GDDAT ;WHAT REGISTER SHOULD BE

```



```

021172 01373/ 001126 001164      MOV      $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
021200 042737 100000 001164      BIC      #^C7777,$TMP0     ;SAVE SPECIFIED BITS
021206 023737 001124 001164      CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
021214 001414      BEQ      72$              ;BR IF OK
021216 013737 001126 001174      MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
021224 042737 077777 001174      BIC      #77777,$TMP4     ;CLEAR THE MASKED BITS
021232 053737 001174 001124      BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
021240 104034      EMT      34
021242 005137 001250      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
021246 000240      72$:    NOP
  
```

;RELEASE THE DRIVE FROM PORT A

```

021250 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)  ;SELECT PORT A
021256 013737 001224 001240      MOV     PORTA, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
021264 012760 000013 000000      MOV     #13, RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

021272 005037 001254      CLR     RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
021276 012737 000012 001122      MOV     #RMDS,$BADDR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
021304 060037 001122      ADD     R0,$BADDR        ;ADD THE I/O BASE ADDRESS
021310 012737 011700 001124      MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
021316 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)  ;SELECT PORT A.
021324 016037 000012 001170      MOV     RMDS(R0),$TMP2    ;GET THE DRIVE STATUS REGISTER FROM PORT A.
021332 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
021340 013737 001170 001164      MOV     $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
021346 042737 100100 001164      BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021354 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)  ;SELECT PORT B.
021362 016037 000012 001172      MOV     RMDS(R0),$TMP3    ;GET THE DRIVE STATUS REGISTER FROM PORT B.
021370 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
021376 013737 001172 001166      MOV     $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
021404 042737 100100 001166      BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
021412 023737 001164 001166      CMP     $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
021420 001006      BNE     74$              ;BR IF NOT
021422 005737 001164      TST     $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
021426 001045      BNE     76$              ;BR IF NOT
021430 104046      EMT     46
021432 000137 021632      JMP     78$              ;BYPASS THE REST OF THE CHECKS
021436 013737 001170 001126 74$:    MOV     $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
021444 013737 001226 001240      MOV     PORTB, PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021452 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)  ;SELECT PORT B.
021460 005737 001164      TST     $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
021464 001414      BEQ     75$              ;BR IF ZERO
021466 013737 001224 001240      MOV     PORTA, PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
021474 013737 001172 001126      MOV     $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
021502 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)  ;SELECT PORT A.
021510 005737 001166      TST     $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
021514 001012      BNE     76$              ;BR IF NOT
021516 012737 177777 001254 75$:    MOV     #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
021524 012760 000011 000000      MOV     #11, RMCS1(R0)   ;CLEAR THE DRIVE
021532 012760 000013 000000      MOV     #13, RMCS1(R0)   ;RELEASE THE DRIVE
021540 104026      EMT     26
021542 013737 001170 001126 76$:    MOV     $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMDS READ
021550 013737 001224 001240      MOV     PORTA, PTNBR     ;CHANGE PORT NUMBER
021556 042737 100000 001126      BIC     #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
021564 023737 001124 001126      CMP     $GDDAT,$BDDAT   ;ALL BITS OK ?
  
```



```

021572 001401 BEQ 77$ ;BR IF OK FROM PORT A.
021574 104007 EMT 7
021576 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
021604 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
021612 042737 10000C 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
021620 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
021626 001401 BEQ 78$ ;BR IF OK
021630 104007 EMT 7
021632 000240 78$: NOP
021634 000004 1$: SCOPE ;LOOP ?
  
```

484
503
504

```

*****
: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.
:
*****
  
```

```

021636 005737 001300 TST15: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
021636 001406 BEQ 2$ ;BR IF NOT
021644 100002 BPL 1$ ;BR IF JUST ENTERED TEST
021646 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
021652 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
021660 012737 021674 001106 2$: MOV #TEST15,$LPADR ;SETUP SCOPE LOOP ADDRESS
021666 012737 021674 001110 MOV #TEST15,$LPERR ;SETUP ERROR LOOP ADDRESS
021674
021674 112737 000015 001102 TEST15: MOVB #15,$STSTNM ;MOVE #15 TO TEST NUMBER
021702 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
021706 012737 000012 001176 MOV #10, $TIMES ;DO 10. ITERATIONS
  
```

505
506

;SEIZE THE DRIVE THROUGH PORT B

```

021714 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
021722 013737 001226 001242 MOV PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
021730 005060 000012 CLR RMDs(R0) ;WRITE RMDs
021734 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
021742 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPREOUT
021750 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
021756 016037 000012 001126 MOV RMDs(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
021764 010037 001122 MOV RO, $BDADR ;RH/RM BASE ADDRESS
021770 062737 000012 001122 ADD #RMDs, $BDADR ;GENERATE BAD REGISTER ADDRESS
021776 005037 001124 CLR $GDDAT ;REGISTER SHOULD BE ZERO
  
```

```

022002 023737 001124 001126      CMP      $GDDAT,$BDDAT      ;IS THE REGISTER ZERO
022010 001403                      BEQ      64$                ;BR IF IT IS
022012 104004                      EMT      4
022014 000137 023234              JMP      1$                ;BYPASS REST OF THE SUBTEST
022020                                64$:
022020 113760 001226 000010          MOV      PORTB, RMCS2(R0)   ;SELECT PORT B
022026 013737 001226 001240          MOV      PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022034 016037 000012 001126          MOV      RMDS(R0), $BDDAT  ;SEE IF SEIZING PORT SEES CORRECT STATUS
022042 042737 020001 001126          BIC      #OM!PIP, $BDDAT   ;CLEAR DONT CARE BITS
022050 012737 011700 001124          MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
022056 013737 001124 001166          MOV      $GDDAT, $TMP1    ;USE GOOD DATA AS A MASK
022064 005137 001166                      STMP1                      ;COMPLEMENT THE EXPECTED STATUS
022070 013737 001126 001164          MOV      $BDDAT, $TMP0    ;SAVE THE ACTUAL STATUS
022076 043737 001166 001164          BIC      $TMP1, $TMP0     ;CLEAR UNWANTED BITS
022104 023737 001124 001164          CMP      $GDDAT, $TMP0    ;ARE THE EXPECTED STATUS BITS SET ?
022112 001401                      BEQ      65$                ;BR IF THEY ARE
022114 104005                      EMT      5
022116 000240                                65$:
                                NOP

                                ;DRIVE CLEAR THROUGH PORT B FIRST

022120 012760 000011 000000          MOV      #11, RMCS1(R0)   ;ISSUE DRIVE CLEAR THROUGH PORT B

                                ;VERIFY THAT DRIVE STILL SEIZED BY PORT B

022126 113760 001224 000010          MOV      PORTA, RMCS2(R0) ;SELECT PORT A
022134 013737 001224 001240          MOV      PORTA, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022142 005037 001250                      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
022146 016037 000012 001126          MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
022154 012737 000012 001122          MOV      #RMDS, $BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022162 060037 001122                      ADD      R0, $BDADR       ;ADD RH/RM BASE ADDRESS
022166 005037 001124                      CLR      $GDDAT          ;WHAT REGISTER SHOULD BE
022172 013737 001126 001164          MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
022200 042737 100000 001164          BIC      #^C77777, $TMP0 ;SAVE SPECIFIED BITS
022206 023737 001124 001164          CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
022214 001414                      BEQ      66$                ;BR IF OK
022216 013737 001126 001174          MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
022224 042737 077777 001174          BIC      #77777, $TMP4   ;CLEAR THE MASKED BITS
022232 053737 001174 001124          BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
022240 104033                      EMT      33
022242 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
022246 000240                                66$:
                                NOP
022250 113760 001226 000010          MOV      PORTB, RMCS2(R0) ;SELECT PORT B
022256 013737 001226 001240          MOV      PORTB, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022264 005037 001250                      CLR      CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
022270 016037 000012 001126          MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
022276 012737 000012 001122          MOV      #RMDS, $BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022304 060037 001122                      ADD      R0, $BDADR       ;ADD RH/RM BASE ADDRESS
022310 012737 011700 001124          MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
022316 013737 001126 001164          MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
022324 042737 100000 001164          BIC      #^C77777, $TMP0 ;SAVE SPECIFIED BITS
022332 023737 001124 001164          CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
022340 001414                      BEQ      68$                ;BR IF OK
022342 013737 001126 001174          MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
022350 042737 077777 001174          BIC      #77777, $TMP4   ;CLEAR THE MASKED BITS
022356 053737 001174 001124          BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
022364 104033                      EMT      33

```

```

022366 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
022372 000240          68$:      NOP

;NOW ISSUE MASSBUS INIT

022374 012760 000040 000010      MOV      #CLR, RMCS2(R0) ;ISSUE MASSBUS INIT

;CONFIRM THAT DRIVE STILL SEIZED BY PORT B

022402 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A
022410 013737 001224 001240      MOV      PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022416 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
022422 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
022430 012737 000012 001122      MOV      #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022436 060037 001122          ADD      R0, $BDADR      ;ADD RH/RM BASE ADDRESS
022442 005037 001124          CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
022446 013737 001126 001164      MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
022454 042737 100000 001164      BIC      #^C77777, $TMP0 ;SAVE SPECIFIED BITS
022462 023737 001124 001164      CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
022470 001414          BEQ      70$            ;BR IF OK
022472 013737 001126 001174      MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
022500 042737 077777 001174      BIC      #77777, $TMP4   ;CLEAR THE MASKED BITS
022506 053737 001174 001124      BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
022514 104034          EMT      34
022516 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
022522 000240          70$:      NOP

022524 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
022532 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022540 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
022544 016037 000012 001126      MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
022552 012737 000012 001122      MOV      #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
022560 060037 001122          ADD      R0, $BDADR      ;ADD RH/RM BASE ADDRESS
022564 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
022572 013737 001126 001164      MOV      $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
022600 042737 100000 001164      BIC      #^C77777, $TMP0 ;SAVE SPECIFIED BITS
022606 023737 001124 001164      CMP      $GDDAT, $TMP0   ;COMPARE THE BITS
022614 001414          BEQ      72$            ;BR IF OK
022616 013737 001126 001174      MOV      $BDDAT, $TMP4   ;COPY 'BAD DATA'
022624 042737 077777 001174      BIC      #77777, $TMP4   ;CLEAR THE MASKED BITS
022632 053737 001174 001124      BIS      $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
022640 104034          EMT      34
022642 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
022646 000240          72$:      NOP

;RELEASE THE DRIVE FROM PORT B

022650 113760 001226 000010      MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
022656 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
022664 012760 000013 000000      MOV      #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

022672 005037 001254          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
022676 012737 000012 001122      MOV      #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
022704 060037 001122          ADD      R0, $BDADR      ;ADD THE I/O BASE ADDRESS
022710 012737 011700 001124      MOV      #MOL!PGM!DPR.DRY.VV, $GDDAT ;COMPARISON CONSTANT
022716 113760 001224 000010      MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
  
```

```

022724 016037 000012 001170      MOV      RMDS(R0), $TMP2      ;GET THE DRIVE STATUS REGISTER FROM PORT A.
022732 042737 024001 001170      BIC      #PIP.WRL!OM, $TMP2      ;CLEAR DONT CARES
022740 013737 001170 001164      MOV      $TMP2, $TMP0      ;COPY IT INTO '$TMP0'
022746 042737 100100 001164      BIC      #ATA!VV, $TMP0      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
022754 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
022762 016037 000012 001172      MOV      RMDS(R0), $TMP3      ;GET THE DRIVE STATUS REGISTER FROM PORT B.
022770 042737 024001 001172      BIC      #PIP.WRL.OM, $TMP3      ;CLEAR DONT CARES
022776 013737 001172 001166      MOV      $TMP3, $TMP1      ;COPY IT INTO '$TMP1'
023004 042737 100100 001166      BIC      #ATA.VV, $TMP1      ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023012 023737 001164 001166      CMP      $TMP0, $TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023020 001006      BNE      74$      ;BR IF NOT
023022 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
023026 001045      BNE      76$      ;BR IF NOT
023030 104046      EMT      46
023032 000137 023232      JMP      78$      ;BYPASS THE REST OF THE CHECKS
023036 013737 001170 001126 74$:  MOV      $TMP2, $BDDAT      ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023044 013737 001226 001240      MOV      PORTB, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023052 113760 001226 000010      MOVVB   PORTB, RMCS2(R0)      ;SELECT PORT B.
023060 005737 001164      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
023064 001414      BEQ      75$      ;BR IF ZERO
023066 013737 001224 001240      MOV      PORTA, PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023074 013737 001172 001126      MOV      $TMP3, $BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
023102 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)      ;SELECT PORT A.
023110 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
023114 001012      BNE      76$      ;BR IF NOT
023116 012737 177777 001254 75$:  MOV      #-1, RELERR      ;SET 'RELEASE ERROR' INDICATOR
023124 012760 000011 000000      MOV      #11, RMCS1(R0)      ;CLEAR THE DRIVE
023132 012760 000013 000000      MOV      #13, RMCS1(R0)      ;RELEASE THE DRIVE
023140 104026      EMT      26
023142 013737 001170 001126 76$:  MOV      $TMP2, $BDDAT      ;LOOK FOR BIT FAILURES WHEN RMDS READ
023150 013737 001224 001240      MOV      PORTA, PTNBR      ;CHANGE PORT NUMBER
023156 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
023164 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;ALL BITS OK ?
023172 001401      BEQ      77$      ;BR IF OK FROM PORT A.
023174 104007      EMT      7
023176 013737 001172 001126 77$:  MOV      $TMP3, $BDDAT      ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
023204 013737 001226 001240      MOV      PORTB, PTNBR      ;CHANGE PORT NUMBER
023212 042737 100000 001126      BIC      #ATA, $BDDAT      ;DON'T CHECK THE ATTN BIT
023220 023737 001124 001126      CMP      $GDDAT, $BDDAT      ;SEE IF READ OK FROM PORT B.
023226 001401      BEQ      78$      ;BR IF OK
023230 104007      EMT      7
023232 000240      NOP      ;
023234 000004      1$:     SCOPE      ;LOOP ?
  
```

507
519
520

```

:*****
:*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.
:*
:*****
  
```

521
531

023236
 023236 005737 001300
 023242 001406
 023244 100002
 023246 000137 003074
 023252 012737 177777 001300
 023260 012737 023274 001106
 023266 012737 023274 001110
 023274
 023274 112737 000016 001102
 023302 012706 001100
 023306 012737 000012 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,\$TSTNM ;MOVE #16 TO TEST NUMBER
 MOV #STACK,\$SP ;LOAD THE STACK POINTER
 MOV #10,\$TIMES ;DO 10. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

023314 113760 001224 000010
 023322 005060 000012
 023326 012760 000011 000000
 023334 012760 000013 000000
 023342 113760 001226 000010
 023350 005060 000012
 023354 012760 000011 000000
 023362 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

023370 113760 001224 000010
 023376 013737 001224 001242
 023404 005760 000000
 023410 113760 001226 000010
 023416 013737 001226 001240
 023424 013737 001226 001244
 023432 016037 000012 001126
 023440 010037 001122
 023444 062737 000012 001122
 023452 005037 001124
 023456 023737 001124 001126
 023464 001403
 023466 104004
 023470 000137 024160
 023474
 023474 113760 001224 000010
 023502 013737 001224 001240
 023510 016037 000012 001126
 023516 042737 020001 001126
 023524 012737 011700 001124
 023532 013737 001124 001166
 023540 005137 001166
 023544 013737 001126 001164
 023552 043737 001166 001164
 023560 023737 001124 001164
 023566 001401
 023570 104005
 023572 000240

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 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\$: MOVB 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

;RELEASE THE DRIVE FROM PORT A

```

023574 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
023602 013737 001224 001240      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
023610 012760 000013 000000      MOV   #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

023616 005037 001254                CLR   RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
023622 012737 000012 001122      MOV   #RMDS,$BDDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
023630 060037 001122                ADD   R0,$BDDADR ;ADD THE I/O BASE ADDRESS
023634 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
023642 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A.
023650 016037 000012 001170      MOV   RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
023656 042737 024001 001170      BIC   #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
023664 013737 001170 001164      MOV   $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
023672 042737 100100 001164      BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023700 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B.
023706 016037 000012 001172      MOV   RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
023714 042737 024001 001172      BIC   #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
023722 013737 001172 001166      MOV   $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
023730 042737 100100 001166      BIC   #ATA.VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
023736 023737 001164 001166      CMP   $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
023744 001006                BNE   66$ ;BR IF NOT
023746 005737 001164                TST   $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
023752 001045                BNE   68$ ;BR IF NOT
023754 104046                EMT   46
023756 000137 024156                JMP   70$ ;BYPASS THE REST OF THE CHECKS
023762 013737 001170 001126 66$: MOV   $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
023770 013737 001226 001240      MOV   PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
023776 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B.
024004 005737 001164                TST   $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
024010 001414                BEQ   67$ ;BR IF ZERO
024012 013737 001224 001240      MOV   PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024020 013737 001172 001126      MOV   $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
024026 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A.
024034 005737 001166                TST   $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
024040 001012                BNE   68$ ;BR IF NOT
024042 012737 177777 001254 67$: MOV   #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
024050 012760 000011 000000      MOV   #11,RMCS1(R0) ;CLEAR THE DRIVE
024056 012760 000013 000000      MOV   #13,RMCS1(R0) ;RELEASE THE DRIVE
024064 104026                EMT   26
024066 013737 001170 001126 68$: MOV   $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
024074 013737 001224 001240      MOV   PORTA,PTNBR ;CHANGE PORT NUMBER
024102 042737 100000 001126      BIC   #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
024110 023737 001124 001126      CMP   $GDDAT,$BDDAT ;ALL BITS OK ?
024116 001401                BEQ   69$ ;BR IF OK FROM PORT A.
024120 104007                EMT   7
024122 013737 001172 001126 69$: MOV   $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
024130 013737 001226 001240      MOV   PORTB,PTNBR ;CHANGE PORT NUMBER
024136 042737 100000 001126      BIC   #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
024144 023737 001124 001126      CMP   $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
024152 001401                BEQ   70$ ;BR IF OK
024154 104007                EMT   7
024156 000240                NOP
024160 000004                1$: SCOPE ;LOOP ?

```

532
544

545

```

*****
*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.
*****
    
```

```

024162
024162 005737 001300
024166 001406
024170 100002
024172 000137 003074
024176 012737 177777 001300
024204 012737 024220 001106
024212 012737 024220 001110
024220
024220 112737 000017 001102
024226 012706 001100
024232 012737 000012 001176
    
```

```

TST17:
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 #TEST17,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST17,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST17:
MOVB #17,$STSTM ;MOVE #17 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #10, $TIMES ;DO 10. ITERATIONS
    
```

546
547

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

024240 113760 001224 000010
024246 005060 000012
024252 012760 000011 000000
024260 012760 000013 000000
024266 113760 001226 000010
024274 005060 000012
024300 012760 000011 000000
024306 012760 000013 000000
    
```

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
CLP 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

```

024314 113760 001226 000010
024322 013737 001226 001242
024330 005760 000000
024334 113760 001224 000010
024342 013737 001224 001240
024350 013737 001224 001244
024356 016037 000012 001126
024364 010037 001122
024370 062737 000012 001122
024376 005037 001124
024402 023737 001124 001126
024410 001403
024412 104004
024414 000137 025104
024420
024420 113760 001226 000010
024426 013737 001226 001240
024434 016037 000012 001126
    
```

```

MOVB PORTB,RMCS2(R0) ;SELECT PORT B
MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
TST RMCS1(R0) ;READ RMCS1
MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
MOV RMDS(R0),$BDDAT ;SEE IF DRIVE SEIZED BY PORT B
MOV RO,$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$:
MOVB PORTB,RMCS2(R0) ;SELECT PORT B
MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMDS(R0),$BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
    
```



```

024442 042737 020001 001126      BIC      #OM,PIP,$BDDAT      ;CLEAR DONT CARE BITS
024450 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
024456 013737 001124 001166      MOV      $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
024464 005137 001166          COM      $TMP1          ;COMPLEMENT THE EXPECTED STATUS
024470 013737 001126 001164      MOV      $BDDAT,$TMP0   ;SAVE THE ACTUAL STATUS
024476 043737 001166 001164      BIC      $TMP1,$TMP0   ;CLEAR UNWANTED BITS
024504 023737 001124 001164      CMP      $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
024512 001401          BEQ      65$           ;BR IF THEY ARE
024514 104005          EMT      5
024516 000240          NOP

;RELEASE THE DRIVE FROM PORT B

024520 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
024526 013737 001226 001240      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
024534 012760 000013 000000      MOV      #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

024542 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
024546 012737 000012 001122      MOV      #RMDS,$BDDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
024554 060037 001122          ADD      R0,$BDDADR    ;ADD THE I/O BASE ADDRESS
024560 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
024566 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
024574 016037 000012 001170      MOV      RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
024602 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
024610 013737 001170 001164      MOV      $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
024616 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024624 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
024632 016037 000012 001172      MOV      RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
024640 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
024646 013737 001172 001166      MOV      $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
024654 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
024662 023737 001164 001166      CMP      $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
024670 001006          BNE      66$           ;BR IF NOT
024672 005737 001164          TST      $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
024676 001045          BNE      68$           ;BR IF NOT
024700 104046          EMT      46
024702 000137 025102          JMP      70$           ;BYPASS THE REST OF THE CHECKS
024706 013737 001170 001126 66$: MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
024714 013737 001226 001240      MOV      PORTB,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024722 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
024730 005737 001164          TST      $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
024734 001414          BEQ      67$           ;BR IF ZERO
024736 013737 001224 001240      MOV      PORTA,PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
024744 013737 001172 001126      MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
024752 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
024760 005737 001166          TST      $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
024764 001012          BNE      68$           ;BR IF NOT
024766 012737 177777 001254 67$: MOV      #-1,RELERR    ;SET 'RELEASE ERROR' INDICATOR
024774 012760 000011 000000      MOV      #11,RMCS1(R0) ;CLEAR THE DRIVE
025002 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
025010 104026          EMT      26
025012 013737 001170 001126 68$: MOV      $TMP2,$BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
025020 013737 001224 001240      MOV      PORTA,PTNBR    ;CHANGE PORT NUMBER
025026 042737 100000 001126      BIC      #ATA,$BDDAT    ;DON'T CHECK THE ATTN BIT
025034 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
  
```



```

025042 001401 BEQ 69$ ;BR IF OK FROM PORT A.
025044 104007 EMT 7
025046 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
025054 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
025062 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
025070 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
025076 001401 BEQ 70$ ;BR IF OK
025100 104007 EMT 7
025102 000240 70$: NOP
025104 000004 1$: SCOPE ;LOOP ?

```

548
504
565

```

:*****
*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.
*
:*****

```

```

025106
025106 005737 001300 TST20: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
025112 001406 BEQ 2$ ;BR IF NOT
025114 100002 BPL 1$ ;BR IF JUST ENTERED TEST
025116 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
025122 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
025130 012737 025144 001106 2$: MOV #TEST20,$LPADR ;SETUP SCOPE LOOP ADDRESS
025136 012737 025144 001110 MOV #TEST20,$LPERR ;SETUP ERROR LOOP ADDRESS
025144
025144 112737 000020 001102 TEST20: MOVB #20,$STNM ;MOVE #20 TO TEST NUMBER
025152 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
025156 012737 000012 001176 MOV #10, $TIMES ;DO 10. ITERATIONS

```

566
583

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

025164 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
025172 005060 000012 CLR RMD5(R0) ;SEIZE THE DRIVE
025176 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
025204 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
025212 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
025220 005060 000012 CLR RMD5(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
025224 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
025232 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE

```

;SEIZE THE DRIVE THROUGH PORT B

```

025240 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
025246 013737 001226 001242 MOV PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS

```

```

025254 005760 000000          TST      RMCS1(R0)      ;READ RMCS1
025260 113760 001224 000010    MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
025266 013737 001224 001240    MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025274 013737 001224 001244    MOV     PORTA,OPPRT    ;'OPPOSITE' PORT ADDRESS
025302 016037 000012 001126    MOV     RMDS(R0),%BDDAT ;SEE IF DRIVE SEIZED BY PORT B
025310 010037 001122          MOV     R0,%BDADR      ;RH/RM BASE ADDRESS
025314 062737 000012 001122    ADD     #RMDS,%BDADR   ;GENERATE BAD REGISTER ADDRESS
025322 005037 001124          CLR     %GDDAT         ;REGISTER SHOULD BE ZERO
025326 023737 001124 001126    CMP     %GDDAT,%BDDAT  ;IS THE REGISTER ZERO
025334 001403          BEQ    64$             ;BR IF IT IS
025336 104004          EMT    4
025340 000137 026152          JMP    1$              ;BYPASS REST OF THE SUBTEST
025344          64$:
025344 113760 001226 000010    MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
025352 013737 001226 001240    MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025360 016037 000012 001126    MOV     RMDS(R0),%BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
025366 042737 020001 001126    BIC     #OM!PIP,%BDDAT ;CLEAR DONT CARE BITS
025374 012737 011700 001124    MOV     #MOL!PGM.DPR!DRY.VV,%GDDAT ;EXPECTED STATUS
025402 013737 001124 001166    MOV     %GDDAT,%STMP1  ;USE GOOD DATA AS A MASK
025410 005137 001166          COM     %STMP1         ;COMPLEMENT THE EXPECTED STATUS
025414 013737 001126 001164    MOV     %BDDAT,%STMP0  ;SAVE THE ACTUAL STATUS
025422 043737 001166 001164    BIC     %STMP1,%STMP0  ;CLEAR UNWANTED BITS
025430 023737 001124 001164    CMP     %GDDAT,%STMP0  ;ARE THE EXPECTED STATUS BITS SET ?
025436 001401          BEQ    65$             ;BR IF THEY ARE
025440 104005          EMT    5
025442 000240          65$:
025444 113760 001224 000010    MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
025452 013737 001224 001240    MOV     PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

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

```

025460 005037 001250          CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
025464 016037 000000 001126    MOV     RMCS1(R0),%BDDAT ;GET CONTENTS OF RMCS1
025472 012737 000000 001122    MOV     #RMCS1,%BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
025500 060037 001122          ADD     R0,%BDADR      ;ADD RH/RM BASE ADDRESS
025504 005037 001124          CLR     %GDDAT         ;WHAT REGISTER SHOULD BE
025510 013737 001126 001164    MOV     %BDDAT,%STMP0  ;MOVE REGISTER CONTENTS TO '%STMP0'
025516 042737 173700 001164    BIC     #^C4077,%STMP0 ;SAVE SPECIFIED BITS
025524 023737 001124 001164    CMP     %GDDAT,%STMP0  ;COMPARE THE BITS
025532 001414          BEQ    66$             ;BR IF OK
025534 013737 001126 001174    MOV     %BDDAT,%STMP4  ;COPY 'BAD DATA'
025542 042737 004077 001174    BIC     #4077,%STMP4   ;CLEAR THE MASKED BITS
025550 053737 001174 001124    BIS     %STMP4,%GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
025556 104010          EMT    10
025560 005137 001250          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
025564 000240          66$:

```

;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT B

```

025566 113760 001226 000010    MOVB    PORTB,RMCS2(R0) ;SELECT PORT B
025574 013737 001226 001240    MOV     PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
025602 012760 000013 000000    MOV     #13,RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT B

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

025610 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
025614 012737 000012 001122 MOV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
025622 060037 001122 ADD R0,$BDADR ;ADD THE I/O BASE ADDRESS
025626 012737 011700 001124 MOV #MOL.PGM.DPR!DRY.VV,$GDDAT ;COMPARISON CONSTANT
025634 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
025642 016037 000012 001170 MOV RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
025650 042737 024001 001170 BIC #PIP.WRL.OM,$TMP2 ;CLEAR DONT CARES
025656 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO 'TMP0'
025664 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025672 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
025700 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
025706 042737 024001 001172 BIC #PIP!WRL.OM,$TMP3 ;CLEAR DONT CARES
025714 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO 'TMP1'
025722 042737 100100 001166 BIC #ATA.VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
025730 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
025736 001006 BNE 68$ ;BR IF NOT
025740 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
025744 001045 BNE 70$ ;BR IF NOT
025746 104046 EMT 46
025750 000137 026150 JMP 72$ ;BYPASS THE REST OF THE CHECKS
025754 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
025762 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
025770 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
025776 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
026002 001414 BEQ 69$ ;BR IF ZERO
026004 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
026012 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
026020 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
026026 005737 001164 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
026032 001012 BNE 70$ ;BR IF NOT
026034 012737 177777 001254 69$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
026042 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
026050 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
026056 104026 EMT 26
026060 013737 001170 001126 70$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
026066 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
026074 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
026102 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
026110 001401 BEQ 71$ ;BR IF OK FROM PORT A.
026112 104007 EMT 7
026114 013737 001172 001126 71$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
026122 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
026130 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
026136 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
026144 001401 BEQ 72$ ;BR IF OK
026146 104007 EMT 7
026150 000240 72$: NOP
026152 000004 1$: SCOPE ;LOOP ?

```

584
600
601

```

*****
*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.
: *
: *****

```

```

026154
026154 005737 001300
026160 001406
026162 100002
026164 000137 003074
026170 012737 177777 001300
026176 012737 026212 001106
026204 012737 026212 001110
026212
026212 112737 000021 001102
026220 012706 001100
026224 012737 000012 001176

```

```

TST21:
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 #TEST21,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST21,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST21:
MOVB #21,$TSTNM ;MOVE #21 TO TEST NUMBER
MOV #STACK,$SP ;LOAD THE STACK POINTER
MOV #10,$TIMES ;DO 10. ITERATIONS

```

602
603

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

026232 113760 001224 000010
026240 005060 000012
026244 012760 000011 000000
026252 012760 000013 000000
026260 113760 001226 000010
026266 005060 000012
026272 012760 000011 000000
026300 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

```

026306 113760 001224 000010
026314 013737 001224 001242
026322 005760 000000
026326 113760 001226 000010
026334 013737 001226 001240
026342 013737 001226 001244
026350 016037 000012 001126
026356 010037 001122
026362 062737 000012 001122
026370 005037 001124
026374 023737 001124 001126
026402 001403
026404 104004
026406 000137 027220
026412
026412 113760 001224 000010
026420 013737 001224 001240
026426 016037 000012 001126
026434 042737 020001 001126
026442 012737 011700 001124
026450 013737 001124 001166

```

```

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$:
MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV RMDS(R0),$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

```

```

026456 005137 001166 COM $TMP1 ;COMPLEMENT THE EXPECTED STATUS
026462 013737 001126 001164 MOV $BDDAT,$TMP0 ;SAVE THE ACTUAL STATUS
026470 043737 001166 001164 BIC $TMP1,$TMP0 ;CLEAR UNWANTED BITS
026476 023737 001124 001164 CMP $GDDAT,$TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
026504 001401 BEQ 65$ ;BR IF THEY ARE
026506 104005 EMT 5
026510 000240 65$: NOP
026512 113760 001226 000010 MOV#B PORTB,RMCS2(R0) ;SELECT PORT B
026520 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
  
```

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

```

026526 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
026532 016037 000000 001126 MOV RMCS1(R0),$BDDAT ;GET CONTENTS OF RMCS1
026540 012737 000000 001122 MOV #RMCS1,$BDDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
026546 060037 001122 ADD R0,$BDDADR ;ADD RH/RM BASE ADDRESS
026552 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
026556 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
026564 042737 173700 001164 BIC #^C4077,$TMP0 ;SAVE SPECIFIED BITS
026572 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
026600 001414 BEQ 66$ ;BR IF OK
026602 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
026610 042737 004077 001174 BIC #4077,$TMP4 ;CLEAR THE MASKED BITS
026616 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
026624 104010 EMT 10
026626 005137 001250 COM ^KERR ;SET THE REGISTER COMPARE ERROR INDICATOR
026632 000240 66$: NOP
  
```

;DRIVE SHOULD RETURN TO NEUTRAL

;RELEASE THE DRIVE FROM PORT A

```

026634 113760 001224 000010 MOV#B PORTA,RMCS2(R0) ;SELECT PORT A
026642 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
026650 012760 000013 000000 MOV #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

026656 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
026662 012737 000012 001122 MOV #RMDS,$BDDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
026670 060037 001122 ADD R0,$BDDADR ;ADD THE I/O BASE ADDRESS
026674 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
026702 113760 001224 000010 MOV#B PORTA,RMCS2(R0) ;SELECT PORT A.
026710 016037 000012 001170 MOV RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
026716 042737 024001 001170 BIC #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
026724 013737 001170 001164 MOV $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
026732 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026740 113760 001226 000010 MOV#B PORTB,RMCS2(R0) ;SELECT PORT B.
026746 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
026754 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
026762 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
026770 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
026776 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
027004 001006 BNE 68$ ;BR IF NOT
027006 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
027012 001045 BNE 70$ ;BR IF NOT
027014 104046 EMT 46
  
```

```

027016 000137 027216      JMP      72$      ;BYPASS THE REST OF THE CHECKS
027022 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
027030 013737 001226 001240      MOV     PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027036 113760 001226 000010      MOVVB   PORTB,RMCS2(R0) ;SELECT PORT B.
027044 005737 001164      TST     $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
027050 001414      BEQ     69$      ;BR IF ZERO
027052 013737 001224 001240      MOV     PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
027060 013737 001172 001126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
027066 113760 001224 000010      MOVVB   PORTA,RMCS2(R0) ;SELECT PORT A.
027074 005737 001166      TST     $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
027100 001012      BNE     70$      ;BR IF NOT
027102 012737 177777 001254 69$:  MOV     #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
027110 012760 000011 000000      MOV     #11,RMCS1(R0) ;CLEAR THE DRIVE
027116 012760 000013 000000      MOV     #13,RMCS1(R0) ;RELEASE THE DRIVE
027124 104026      EMT     26
027126 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
027134 013737 001224 001240      MOV     PORTA,PTNBR ;CHANGE PORT NUMBER
027142 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
027150 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
027156 001401      BEQ     71$      ;BR IF OK FROM PORT A.
027160 104007      EMT     7
027162 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
027170 013737 001226 001240      MOV     PORTB,PTNBR ;CHANGE PORT NUMBER
027176 042737 100000 001126      BIC     #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
027204 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
027212 001401      BEQ     72$      ;BR IF OK
027214 104007      EMT     7
027216 000240      NOP
027220 000004      1$:    SCOPE      ;LOOP ?
  
```

604
619
620

```

*****
*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.
*****
  
```

```

027222 005737 001300      TST22: TST     KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
027222 001406      BEQ     2$      ;BR IF NOT
027230 100002      BPL     1$      ;BR IF JUST ENTERED TEST
027232 000137 003074      JMP     EXEC     ;RETURN & GET NEXT TEST NUMBER
027236 012737 177777 001300 1$:  MOV     #-1,KYBCTL ;SET SINGLE TEST INDICATOR
027244 012737 027260 001106 2$:  MOV     #TEST22,$LPADR ;SETUP SCOPE LOOP ADDRESS
027252 012737 027260 001110      MOV     #TEST22,$LPERR ;SETUP ERROR LOOP ADDRESS
027260
027260 112737 000022 001102 TEST22: MOVVB   #22,$STNM   ;MOVE #22 TO TEST NUMBER
027266 012706 001100      MOV     #STACK,SP ;LOAD THE STACK POINTER
  
```

621
675

027272 012737 000012 001176 MOV #10.,\$TIMES ;;DO 10. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

027300 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
 027306 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE
 027312 012760 000011 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
 027320 012760 000013 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
 027326 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
 027334 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
 027340 012760 000011 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
 027346 012760 000013 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE

;SELECT DRIVE OTHER THAN THAT BEING TESTED

027354 113760 001230 000010 MOVB PORTC,RMCS2(R0) ;SELECT DRIVE NOT BEING TESTED
 027362 013737 001224 001242 MOV PORTA,SEIZPT ;'SEIZED' PORT ADDRESS

;WRITE THE DRIVE'S ATTENTION BIT

027370 013760 001236 000016 MOV ASR1,RMAS(R0) ;WRITE THE ATTENTION BIT OF THE DRIVE BEING TESTED
 027376 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
 027404 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

027412 005760 000012 TST RMDS(R0) ;SEE THE REGISTER THROUGH PORT A ?
 027416 001014 BNE 1\$;BR IF YES
 027420 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
 027426 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 027434 005760 000012 TST RMDS(R0) ;SEE REGISTER THROUGH PORT B ?
 027440 001021 BNE 2\$;BR IF YES
 027442 104042 EMT 42
 027444 000137 031274 JMP 4\$;BYPASS REST OF TEST
 027450 1\$:
 027450 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
 027456 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
 027464 005760 000012 TST RMDS(R0) ;REGISTER SHOULD BE ZERO THROUGH PORT B
 027470 001002 BNE +6 ;BR IF STATUS REG IS NOT ZERO
 027472 000137 030374 JMP 3\$;STATUS REG IS ZERO
 027476 104043 EMT 43
 027500 000137 031274 JMP 4\$;BYPASS REST OF TEST

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

027504 2\$:
 027504 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
 027510 016037 000012 001126 MOV RMDS(R0),\$BDDAT ;GET CONTENTS OF RMDS
 027516 012737 000012 001122 MOV #RMDS,\$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
 027524 060037 001122 ADD R0,\$BDADR ;ADD RH/RM BASE ADDRESS
 027530 012737 011700 001124 MOV #MOL!PGM!DPR!DRY!VV,\$GDDAT ;WHAT REGISTER SHOULD BE
 027536 013737 001126 001164 MOV \$BDDAT,\$TMP0 ;MOVE REGISTER CONTENTS TO '\$TMP0'
 027544 042737 106077 001164 BIC #^C71700,\$TMP0 ;SAVE SPECIFIED BITS
 027552 023737 001124 001164 CMP \$GDDAT,\$TMP0 ;COMPARE THE BITS
 027560 001414 BEQ 64\$;BR IF OK
 027562 013737 001126 001174 MOV \$BDDAT,\$TMP4 ;COPY 'BAD DATA'


```

027570 042737 071700 001174      BIC      #71700,$TMP4      ;CLEAR THE MASKED BITS
027576 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
027604 104010                      EMT      10
027606 005137 001250                      COM      CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
027612 000240                      64$:    NOP
027614 013737 001226 001242      MOV      PORTB,SEIZPT    ;ADDRESS FOR ERROR MESSAGE
027622 013737 001224 001244      MOV      PORTA,OPPRT    ;SAME AS ABOVE
  
```

;RELEASE THE DRIVE FROM PORT B

```

027630 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B
027636 013737 001226 001240      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027644 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

```

027652 005037 001254                      CLR      RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
027656 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
027664 012737 000012 001122      MOV      #RMDS,$BDADR   ;REGISTER ADDRESS INCREMENT
027672 060037 001122                      ADD      R0,$BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
027676 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
027704 013737 001224 001240      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027712 016037 000012 001164      MOV      RMDS(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
027720 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
027726 013737 001226 001240      MOV      PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
027734 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;DRIVE STATUS FROM PORT B
027742 001404                      BEQ      66$            ;BR IF STATUS FROM PORT B ZERO
027744 005737 001164                      TST      $TMP0          ;IS STATUS FROM PORT A ZERO ?
027750 001401                      BEQ      66$            ;BR IF ZERO
027752 104044                      EMT      44
027754 013737 001164 001126      66$:    MOV      $TMP0,$BDDAT    ;CHECK STATUS FROM PORT A
027762 013737 001224 001240      MOV      PORTA,PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
027770 023737 001124 001126      CMP      $GDDAT,$BDDAT  ;COMPARE WITH CONSTANT
027776 001401                      BEQ      67$            ;BR IF OK
030000 104027                      EMT      27
030002 000240                      67$:    NOP
  
```

;RELEASE THE DRIVE FROM PORT A

```

030004 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
030012 013737 001224 001240      MOV      PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030020 012760 000013 000000      MOV      #13,RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT A
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

030026 005037 001254                      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
030032 012737 000012 001122      MOV      #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
030040 060037 001122                      ADD      R0,$BDADR      ;ADD THE I/O BASE ADDRESS
030044 012737 011700 001124      MOV      #MOI!PGM!DPR!DRY!VV,$GDDAT    ;COMPARISON CONSTANT
030052 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
030060 016037 000012 001170      MOV      RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
030066 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
030074 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO 'TMP0'
030102 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030110 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
030116 016037 000012 001172      MOV      RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B
030124 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
  
```



```

030132 013737 001172 001166      MOV      $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
030140 042737 100100 001166      BIC      #ATA,VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
030146 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
030154 001006                BNE      68$              ;BR IF NOT
030156 005737 001164                TST      $TMP0            ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
030162 001045                BNE      70$              ;BR IF NOT
030164 104046                EMT      46
030166 000137 030366                JMP      72$              ;BYPASS THE REST OF THE CHECKS
030172 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT     ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
030200 013737 001226 001240        MOV      PORTB,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030206 113760 001226 000010        MOVVB   PORTB,RMCS2(R0)   ;SELECT PORT B.
030214 005737 001164                TST      $TMP0            ;SEE IF STATUS EQ 0 FROM PORT A.
030220 001414                BEQ      69$              ;BR IF ZERO
030222 013737 001224 001240        MOV      PORTA,PTNBR      ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
030230 013737 001172 001126        MOV      $TMP3,$BDDAT     ;'BAD DATA' FOR ERROR TYPE OUT
030236 113760 001224 000010        MOVVB   PORTA,RMCS2(R0)   ;SELECT PORT A.
030244 005737 001166                TST      $TMP1            ;SEE IF STATUS EQ ZERO FROM PORT B.
030250 001012                BNE      70$              ;BR IF NOT
030252 012737 177777 001254 69$:  MOV      #-1,RELERR       ;SET 'RELEASE ERROR' INDICATOR
030260 012760 000011 000000        MOV      #11,RMCS1(R0)    ;CLEAR THE DRIVE
030266 012760 000013 000000        MOV      #13,RMCS1(R0)    ;RELEASE THE DRIVE
030274 104026                EMT      26
030276 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT     ;LOOK FOR BIT FAILURES WHEN RMD5 READ
030304 013737 001224 001240        MOV      PORTA,PTNBR      ;CHANGE PORT NUMBER
030312 042737 100000 001126        BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
030320 023737 001124 001126        CMP      $GDDAT,$BDDAT    ;ALL BITS OK ?
030326 001401                BEQ      71$              ;BR IF OK FROM PORT A.
030330 104007                EMT      7
030332 013737 001172 001126 71$:  MOV      $TMP3,$BDDAT     ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
030340 013737 001226 001240        MOV      PORTB,PTNBR      ;CHANGE PORT NUMBER
030346 042737 100000 001126        BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
030354 023737 001124 001126        CMP      $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
030362 001401                BEQ      72$              ;BR IF OK
030364 104007                EMT      7
030366 000240                NOP
030370 000137 031274                JMP      4$

```

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

```

030374 113760 001224 000010 3$:  MOVVB   PORTA,RMCS2(R0)   ;SELECT PORT A
030402 013737 001224 001240        MOV      PORTA,PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030410 005037 001250                CLR      CKERR             ;CLEAR THE 'CHECK ERROR' INDICATOR
030414 016037 000012 001126        MOV      RMD5(R0),$BDDAT   ;GET CONTENTS OF RMD5
030422 012737 000012 001122        MOV      #RMD5,$BDDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
030430 060037 001122                ADD      R0,$BDDADR        ;ADD RH/RM BASE ADDRESS
030434 012737 011700 001124        MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;WHAT REGISTER SHOULD BE
030442 013737 001126 001164        MOV      $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
030450 042737 106077 001164        BIC      #^C71700,$TMP0   ;SAVE SPECIFIED BITS
030456 023737 001124 001164        CMP      $GDDAT,$TMP0     ;COMPARE THE BITS
030464 001414                BEQ      73$              ;BR IF OK
030466 013737 001126 001174        MOV      $BDDAT,$TMP4     ;COPY 'BAD DATA'
030474 042737 071700 001174        BIC      #71700,$TMP4     ;CLEAR THE MASKED BITS
030502 053737 001174 001124        BIS      $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
030510 104010                EMT      10
030512 005137 001250                COM      CKERR
030516 000240                NOP
73$:

```

```
030520 013737 001224 001242      MOV   PORTA,SEIZPT  ;ADDRESS FOR ERROR MESSAGE
030526 013737 001226 001244      MOV   PORTB,OPPRT  ;SAME AS ABOVE
```

;RELEASE THE DRIVE FROM PORT A

```
030534 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
030542 013737 001224 001240      MOV   PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030550 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

```
030556 005037 001254                CLR   RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
030562 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
030570 012737 000012 001122      MOV   #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
030576 060037 001122                ADD   R0,$BDADR     ;REGISTER BASE ADDRESS FOR TYPEOUT
030602 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
030610 013737 001226 001240      MOV   PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030616 016037 000012 001164      MOV   RMDS(R0),$TMP0 ;READ STATUS REGISTER FROM PORT B
030624 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
030632 013737 001224 001240      MOV   PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030640 016037 000012 001126      MOV   RMDS(R0),$BDAT ;DRIVE STATUS FROM PORT A
030646 001404                BEQ   75$           ;BR IF STATUS FROM PORT A ZERO
030650 005737 001164                TST   $TMP0         ;IS STATUS FROM PORT B ZERO ?
030654 001401                BEQ   75$           ;BR IF ZERO
030656 104044                EMT   44
030660 013737 001164 001126 75$:  MOV   $TMP0,$BDAT   ;CHECK STATUS FROM PORT B
030666 013737 001226 001240      MOV   PORTB,PTNBR  ;CHANGE PORT ADDRESS FOR TYPEOUT
030674 023737 001124 001126      CMP   $GDDAT,$BDAT ;COMPARE WITH CONSTANT
030702 001401                BEQ   76$           ;BR IF OK
030704 104027                EMT   27
030706 000240                76$:  NOP
```

;RELEASE THE DRIVE FROM PORT B

```
030710 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
030716 013737 001226 001240      MOV   PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
030724 012760 000013 000000      MOV   #13,RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
030732 005037 001254                CLR   RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
030736 012737 000012 001122      MOV   #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
030744 060037 001122                ADD   R0,$BDADR     ;ADD THE I/O BASE ADDRESS
030750 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
030756 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT A
030764 016037 000012 001170      MOV   RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A
030772 042737 024001 001170      BIC   #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
031000 013737 001170 001164      MOV   $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
031006 042737 100100 001164      BIC   #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031014 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT B
031022 016037 000012 001172      MOV   RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B
031030 042737 024001 001172      BIC   #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
031036 013737 001172 001166      MOV   $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
031044 042737 100100 001166      BIC   #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031052 023737 001164 001166      CMP   $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031060 001006                BNE   77$           ;BR IF NOT
031062 005737 001164                TST   $TMP0         ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
```

```

031066 001045 BNE 79$ ;BR IF NOT
031070 104046 EMT 46
031072 000137 031272 JMP 81$ ;BYPASS THE REST OF THE CHECKS
031076 013737 001170 001126 77$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031104 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031112 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
031120 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
031124 001414 BEQ 78$ ;BR IF ZERO
031126 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031134 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
031142 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
031150 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
031154 001012 BNE 79$ ;BR IF NOT
031156 012737 177777 001254 78$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
031164 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
031172 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
031200 104026 EMT 26
031202 013737 001170 001126 79$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
031210 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
031216 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
031224 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
031232 001401 BEQ 80$ ;BR IF OK FROM PORT A.
031234 104007 EMT 7
031236 013737 001172 001126 80$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
031244 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
031252 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
031260 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
031266 001401 BEQ 81$ ;BR IF OK
031270 104007 EMT 7
031272 000240 81$: NOP
031274 000004 4$: SCOPE ;LOOP ?
  
```

676
688
689

```

*****
*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.
*****
  
```

```

031276 005737 001300 TST23: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
031276 001406 BEQ 2$ ;BR IF NOT
031302 100002 BPL 1$ ;BR IF JUST ENTERED TEST
031306 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
031312 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
031320 012737 031334 001106 2$: MOV #TEST23,$LPADR ;SETUP SCOPE LOOP ADDRESS
031326 012737 031334 001110 MOV #TEST23,$LPERR ;SETUP ERROR LOOP ADDRESS
031334 TEST23: MOVB #23,$STSTM ;MOVE #23 TO TEST NUMBER
031334 112737 000023 001102 MOV #STACK,SP ;LOAD THE STACK POINTER
031342 012706 001100 MOV #10, $TIMES ;DO 10. ITERATIONS
031346 012737 000012 001176
  
```

690
705

:CLEAR ATTENTION BITS FOR BOTH PORTS

```

031354 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
031362 005060 000012              CLR   RMDS(R0)         ;SEIZE THE DRIVE
031366 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
031374 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
031402 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
031410 005060 000012              CLR   RMDS(R0)         ;SEIZE THE DRIVE THROUGH PORT 'B'
031414 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
031422 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
031430 113760 001230 000010      MOVB  PORTC, RMCS2(R0) ;SELECT DRIVE NOT BEING TESTED
    
```

:WRITE ALL ATTENTION BITS EXCEPT BIT FOR DRIVE UNDER TEST

```

031436 013737 001236 001164      MOV   ASR1, $TMP0     ;STORE ATTN BIT FOR PORT A
031444 005137 001164              COM   $TMP0           ;COMPLEMENT IT
031450 013760 001164 000016      MOV   $TMP0, RMAS(R0) ;WRITE THE ATTN REGISTER
    
```

:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

031456 005037 001254              CLR   RELERR          ;CLEAR THE 'RELEASE ERROR' INDICATOR
031462 012737 000012 001122      MOV   #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
031470 060037 001122              ADD   R0, $BDADR      ;ADD THE I/O BASE ADDRESS
031474 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
031502 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
031510 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
031516 042737 024001 001170      BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
031524 013737 001170 001164      MOV   $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
031532 042737 100100 001164      BIC   #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031540 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
031546 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
031554 042737 024001 001172      BIC   #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
031562 013737 001172 001166      MOV   $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
031570 042737 100100 001166      BIC   #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
031576 023737 001164 001166      CMP   $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
031604 001006              BNE   64$            ;BR IF NOT
031606 005737 001164              TST   $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
031612 001045              BNE   66$            ;BR IF NOT
031614 104046              EMT   46
031616 000137 032016              JMP   68$            ;BYPASS THE REST OF THE CHECKS
031622 013737 001170 001126 64$:  MOV   $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
031630 013737 001226 001240      MOV   PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031636 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
031644 005737 001164              TST   $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
031650 001414              BEQ   65$            ;BR IF ZERO
031652 013737 001224 001240      MOV   PORTA, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
031660 013737 001172 001126      MOV   $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
031666 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
031674 005737 001166              TST   $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
031700 001012              BNE   66$            ;BR IF NOT
031702 012737 177777 001254 65$:  MOV   #-1, RELERR     ;SET 'RELEASE ERROR' INDICATOR
031710 012760 000011 000000      MOV   #11, RMCS1(R0)  ;CLEAR THE DRIVE
031716 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
031724 104021              EMT   21
031726 013737 001170 001126 66$:  MOV   $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
    
```

*23 INHIBIT SEIZE BY RMAS TEST

```

031734 013737 001224 001240      MOV   PORTA,PTNBR      ;CHANGE PORT NUMBER
031742 042737 100000 001126      BIC   #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
031750 023737 001124 001126      CMP   $GDDAT,$BDDAT   ;ALL BITS OK ?
031756 001401                      BEQ   67$              ;BR IF OK FROM PORT A.
031760 104007                      EMT   7
031762 013737 001172 001126 67$: MOV   $TMP3,$BDDAT     ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
031770 013737 001226 001240      MOV   PORTB,PTNBR     ;CHANGE PORT NUMBER
031776 042737 100000 001126      BIC   #ATA,$BDDAT     ;DON'T CHECK THE ATTN BIT
032004 023737 001124 001126      CMP   $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
032012 001401                      BEQ   68$              ;BR IF OK
032014 104007                      EMT   7
032016 000240 68$: NOP
032020 000004                      SCOPE                  ;LOOP ?

```

706
725
726

```

:*****
:*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.
:*
:*****

```

```

032022 005737 001300      TST   KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
032026 001406                      BEQ   2$              ;BR IF NOT
032030 100002                      BPL   1$              ;BR IF JUST ENTERED TEST
032032 000137 003074      JMP   EXEC            ;RETURN & GET NEXT TEST NUMBER
032036 012737 177777 001300 1$: MOV   #-1,KYBCTL     ;SET SINGLE TEST INDICATOR
032044 012737 032060 001106 2$: MOV   #TEST24,$LPADR ;SETUP SCOPE LOOP ADDRESS
032052 012737 032060 001110      MOV   #TEST24,$LPERR ;SETUP ERROR LOOP ADDRESS
032060                      TEST24:
032060 112737 000024 001102      MOVB  #24,$STSTM      ;MOVE #24 TO TEST NUMBER
032066 012706 001100      MOV   #STACK,SP      ;LOAD THE STACK POINTER
032072 012737 000012 001176      MOV   #10,$TIMES     ;;DO 10. ITERATIONS

```

727
756

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

032100 113760 001224 000010      MOVB  PORTA,RMCS2(R0) ;SELECT PORT #A
032106 005060 000012      CLR   RMDs(R0)       ;SEIZE THE DRIVE
032112 012760 000011 000000      MOV   #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR
032120 012760 000013 000000      MOV   #13,RMCS1(R0)  ;RELEASE THE DRIVE
032126 113760 001226 000010      MOVB  PORTB,RMCS2(R0) ;SELECT PORT #B
032134 005060 000012      CLR   RMDs(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
032140 012760 000011 000000      MOV   #11,RMCS1(R0)  ;ISSUE DRIVE CLEAR

```

```

032146 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
;SEIZE THE DRIVE THROUGH PORT B

032154 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
032162 013737 001226 001242      MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
032170 005060 000012                CLR      RMDS(R0) ;WRITE RMDS
032174 013737 001224 001244      MOV      PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS
032202 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
032210 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET PORT REQUEST

032216 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

032222 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
032230 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032236 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

032244 005037 001254                CLR      RELERR ;CLEAR 'RELEASE ERROR' INDICATOR
032250 012737 111700 001124      MOV      #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
032256 012737 000012 001122      MOV      #RMDS,$BDADR ;REGISTER ADDRESS INCREMENT
032264 060037 001122                ADD      R0,$BDADR ;REGISTER BASE ADDRESS FOR TYPEOUT
032270 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
032276 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032304 016037 000012 001164      MOV      RMDS(R0),$TMP0 ;READ STATUS REGISTER FROM PORT A
032312 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B
032320 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032326 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;DRIVE STATUS FROM PORT B
032334 001404                BEQ      66$ ;BR IF STATUS FROM PORT B ZERO
032336 005737 001164                TST      $TMP0 ;IS STATUS FROM PORT A ZERO ?
032342 001401                BEQ      66$ ;BR IF ZERO
032344 104031                EMT      31
032346 013737 001164 001126 66$: MOV      $TMP0,$BDDAT ;CHECK STATUS FROM PORT A
032354 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT ADDRESS FOR TYPEOUT
032362 023737 001124 001126      CMP      $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
032370 001401                BEQ      67$ ;BR IF OK
032372 104027                EMT      27
032374 000240                NOP
032376 113760 001226 000010 67$: MOV      PORTB,RMCS2(R0) ;SELECT PORT B
032404 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032412 005037 001250                CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
032416 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
032424 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032432 060037 001122                ADD      R0,$BDADR ;ADD RH/RM BASE ADDRESS
032436 005037 001124                CLR      $GDDAT ;WHAT REGISTER SHOULD BE
032442 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
032450 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
032456 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
032464 001414                BEQ      68$ ;BR IF OK
032466 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
  
```

```

032474 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
032502 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
032510 104016                      EMT      16
032512 005137 001250                      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
032516 000240                      NOP
68$: 032520 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
032526 013737 001224 001240      MOV      PORTA,PINBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032534 005037 001250                      CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
032540 016037 000012 001126      MOV      RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
032546 012737 000012 001122      MOV      #RMD5,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
032554 060037 001122                      ADD      R0,$BDADR      ;ADD RH/RM BASE ADDRESS
032560 012737 100000 001124      MOV      #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
032566 013737 001126 001164      MOV      $BDDAT,$TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
032574 042737 077777 001164      BIC      #^CATA,$TMP0  ;SAVE SPECIFIED BITS
032602 023737 001124 001164      CMP      $GDDAT,$TMP0  ;COMPARE THE BITS
032610 001414                      BEQ      70$           ;BR IF OK
032612 013737 001126 001174      MOV      $BDDAT,$TMP4  ;COPY 'BAD DATA'
032620 042737 100000 001174      BIC      #ATA,$TMP4    ;CLEAR THE MASKED BITS
032626 053737 001174 001124      BIS      $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
032634 104016                      EMT      16
032636 005137 001250                      COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
0326 2 000240                      NOP
70$:

;RELEASE THE DRIVE FROM PORT A

032644 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
032652 013737 001224 001240      MOV      PORTA,PINBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
032660 012760 000013 000000      MOV      #13,RMCS1(R0)  ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

032666 005037 001254                      CLR      RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
032672 012737 000012 001122      MOV      #RMD5,$BDADR   ;FORM THE ADDRESS OF RMD5 FOR TYPEOUT
032700 060037 001122                      ADD      R0,$BDADR      ;ADD THE I/O BASE ADDRESS
032704 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
032712 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A.
032720 016037 000012 001170      MOV      RMD5(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
032726 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
032734 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
032742 042737 100100 001164      BIC      #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
032750 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B.
032756 016037 000012 001172      MOV      RMD5(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
032764 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
032772 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
033000 042737 100100 001166      BIC      #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
033006 023737 001164 001166      CMP      $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
033014 001006                      BNE      72$           ;BR IF NOT
033016 005737 001164                      TST      $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
033022 001045                      BNE      74$           ;BR IF NOT
033024 104046                      EMT      46
033026 000137 033212                      JMP      76$           ;BYPASS THE REST OF THE CHECKS
72$: 033032 013737 001170 001126      MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
033040 013737 001226 001240      MOV      PORTB,PINBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
033046 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B.
033054 005737 001164                      TST      $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
033060 001414                      BEQ      73$           ;BR IF ZERO
033062 013737 001224 001240      MOV      PORTA,PINBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
    
```



```

033070 013737 001172 001126      MOV      $TMP3,$BDDAT      ;'BAD DATA' FOR ERROR TYPE OUT
033076 113760 001224 000010      MOVVB   PORTA, RMCS2(R0)  ;SELECT PORT A.
033104 005737 001166                TST     $TMP1              ;SEE IF STATJS EQ ZERO FROM PORT B.
033110 001012                BNE     74$                ;BR IF NOT
033112 012737 177777 001254 73$:  MOV     #-1, RELERR        ;SET 'RELEASE ERROR' INDICATOR
033120 012760 000011 000000      MOV     #11, RMCS1(R0)    ;CLEAR THE DRIVE
033126 012760 000013 000000      MOV     #13, RMCS1(R0)    ;RELEASE THE DRIVE
033134 104026                EMT     26
033136 013737 001170 001126 74$:  MOV     $TMP2,$BDDAT      ;LOOK FOR BIT FAILURES WHEN RMD5 READ
033144 013737 001224 001240      MOV     PORTA, PTNBR      ;CHANGE PORT NUMBER
033152 023737 001124 001126      CMP     $GDDAT,$BDDAT    ;ALL BITS OK ?
033160 001401                BEQ     75$                ;BR IF OK FROM PORT A.
033162 104007                EMT     7
033164 013737 001172 001126 75$:  MOV     $TMP3,$BDDAT      ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
033172 013737 001226 001240      MOV     PORTB, PTNBR     ;CHANGE PORT NUMBER
033200 023737 001124 001126      CMP     $GDDAT,$BDDAT    ;SEE IF READ OK FROM PORT B.
033206 001401                BEQ     76$                ;BR IF OK
033210 104007                EMT     7
033212 000240                NOP
033214 000004                1$:    SCOPE                ;LOOP ?

```

757
776
777

```

*****
*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 RMD5.
*
*B. WRITE 0'S INTO RMD5 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.
*****

```

```

033216 005737 001300      TST     KYBCTL            ;PERFORMING ONLY SINGLE TEST ?
033222 001406                BEQ     2$                ;BR IF NOT
033224 100002                BPL     1$                ;BR IF JUST ENTERED TEST
033226 000137 003074      JMP     EXEC              ;RETURN & GET NEXT TEST NUMBER
033232 012737 177777 001300 1$:  MOV     #-1, KYBCTL        ;SET SINGLE TEST INDICATOR
033240 012737 033254 001106 2$:  MOV     #TEST25,$LPADR    ;SETUP SCOPE LOOP ADDRESS
033246 012737 033254 001110      MOV     #TEST25,$LPERR   ;SETUP ERROR LOOP ADDRESS
033254                TEST25:
033254 112737 000025 001102      MOVVB  #25,$STNM         ;MOVE #25 TO TEST NUMBER
033262 012706 001100      MOV     #STACK, SP       ;LOAD THE STACK POINTER
033266 012737 000012 001176      MOV     #10., $TIMES     ;DO 10. ITERATIONS

```

778
779

;CLEAR ATTENTION BITS FOR BOTH PORTS


```

033274 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
033302 005060 000012 000000      CLR   RMDS(R0)        ;SEIZE THE DRIVE
033306 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
033314 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
033322 113760 001224 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
033330 005060 000012 000000      CLR   RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
033334 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
033342 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT A

033350 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
033356 013737 001224 001242      MOV   PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
033364 005060 000012 000000      CLR   RMDS(R0)        ;WRITE RMDS
033370 013737 001226 001244      MOV   PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
033376 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
033404 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;SET PORT REQUEST

033412 005060 000012 000000      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

033416 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
033424 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033432 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

033440 005037 001254 000000      CLR   RELERR          ;CLEAR 'RELEASE ERROR' INDICATOR
033444 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
033452 012737 000012 001122      MOV   #RMDS, $BDADR   ;REGISTER ADDRESS INCREMENT
033460 060037 001122 000000      ADD   R0, $BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
033464 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
033472 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033500 016037 000012 001164      MOV   RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT B
033506 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
033514 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033522 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT A
033530 001404 000000 000000      BEQ   66$             ;BR IF STATUS FROM PORT A ZERO
033532 005737 001164 000000      TST   $TMP0           ;IS STATUS FROM PORT B ZERO ?
033536 001401 000000 000000      BEQ   66$             ;BR IF ZERO
033540 104031 000000 000000      EMT   31
033542 013737 001164 001126 66$: MOV   $TMP0, $BDDAT    ;CHECK STATUS FROM PORT B
033550 013737 001226 001240      MOV   PORTB, PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
033556 023737 001124 001126      CMP   $GDDAT, $BDDAT  ;COMPARE WITH CONSTANT
033564 001401 000000 000000      BEQ   67$             ;BR IF OK
033566 104027 000000 000000      EMT   27
033570 000240 000000 000000 67$: NOP
033572 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
033600 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033606 005037 001250 000000      CLR   CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
033612 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
033620 012737 000012 001122      MOV   #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
  
```

```

033626 060037 001122      ADD    R0,$BDADR      ;ADD RH/RM BASE ADDRESS
033632 005037 001124      CLR    $GDDAT        ;WHAT REGISTER SHOULD BE
033636 013737 001126 001164  MOV    $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
033644 042737 077777 001164  BIC    #^CATA,$STMP0 ;SAVE SPECIFIED BITS
033652 023737 001124 001164  CMP    $GDDAT,$STMP0 ;COMPARE THE BITS
033660 001414      BEQ    68$           ;BR IF OK
033662 013737 001126 001174  MOV    $BDDAT,$STMP4 ;COPY 'BAD DATA'
033670 042737 100000 001174  BIC    #ATA,$STMP4   ;CLEAR THE MASKED BITS
033676 053737 001174 001124  BIS    $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
033704 104016      EMT    16
033706 005137 001250      COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
033712 000240      NOP
68$:
033714 113760 001226 000010  MOV    PORTB, RMCS2(R0) ;SELECT PORT B
033722 013737 001226 001240  MOV    PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
033730 005037 001250      CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
033734 016037 000012 001126  MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
033742 012737 000012 001122  MOV    #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
033750 060037 001122      ADD    R0,$BDADR     ;ADD RH/RM BASE ADDRESS
033754 012737 100000 001124  MOV    #ATA,$GDDAT   ;WHAT REGISTER SHOULD BE
033762 013737 001126 001164  MOV    $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO '$STMP0'
033770 042737 077777 001164  BIC    #^CATA,$STMP0 ;SAVE SPECIFIED BITS
033776 023737 001124 001164  CMP    $GDDAT,$STMP0 ;COMPARE THE BITS
034004 001414      BEQ    70$           ;BR IF OK
034006 013737 001126 001174  MOV    $BDDAT,$STMP4 ;COPY 'BAD DATA'
034014 042737 100000 001174  BIC    #ATA,$STMP4   ;CLEAR THE MASKED BITS
034022 053737 001174 001124  BIS    $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
034030 104016      EMT    16
034032 005137 001250      COM    CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
034036 000240      NOP
70$:

```

:RELEASE THE DRIVE FROM PORT B

```

034040 113760 001226 000010  MOV    PORTB, RMCS2(R0) ;SELECT PORT B
034046 013737 001226 001240  MOV    PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034054 012760 000013 000000  MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B

```

:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

034062 005037 001254      CLR    REIERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
034066 012737 000012 001122  MOV    #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
034074 060037 001122      ADD    R0,$BDADR     ;ADD THE I/O BASE ADDRESS
034100 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
034106 113760 001224 000010  MOV    PORTA, RMCS2(R0) ;SELECT PORT A.
034114 016037 000012 001170  MOV    RMDS(R0), $STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
034122 042737 024001 001170  BIC    #PIP!WRL!OM,$STMP2 ;CLEAR DONT CARES
034130 013737 001170 001164  MOV    $STMP2,$STMP0 ;COPY IT INTO '$STMP0'
034136 042737 100100 001164  BIC    #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034144 113760 001226 000010  MOV    PORTB, RMCS2(R0) ;SELECT PORT B.
034152 016037 000012 001172  MOV    RMDS(R0), $STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
034160 042737 024001 001172  BIC    #PIP!WRL!OM,$STMP3 ;CLEAR DONT CARES
034166 013737 001172 001166  MOV    $STMP3,$STMP1 ;COPY IT INTO '$STMP1'
034174 042737 100100 001166  BIC    #ATA!VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
034202 023737 001164 001166  CMP    $STMP0,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
034210 001006      BNE    72$           ;BR IF NOT
034212 005737 001164      TST    $STMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
034216 001045      BNE    74$           ;BR IF NOT
034220 104046      EMT    46

```

```

034222 000137 034406      JMP 76$ ;BYPASS THE REST OF THE CHECKS
034226 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
034234 013737 001226 001240      MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034242 113760 001226 000010      MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
034250 005737 001164      TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
034254 001414      BEQ 73$ ;BR IF ZERO
034256 013737 001224 001240      MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
034264 013737 001172 001126      MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
034272 113760 001224 000010      MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
034300 005737 001166      TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
034304 001012      BNE 74$ ;BR IF NOT
034306 012737 177777 001254 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
034314 012760 000011 000000      MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
034322 012760 000013 000000      MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
034330 104026      EMT 26
034332 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
034340 013737 001224 001240      MOV PORTA,PTNBR ;CHANGE PORT NUMBER
034346 023737 001124 001126      CMP $GDDAT,$BDDAT ;ALL BITS OK ?
034354 001401      BEQ 75$ ;BR IF OK FROM PORT A.
034356 104007      EMT 7
034360 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
034366 013737 001226 001240      MOV PORTB,PTNBR ;CHANGE PORT NUMBER
034374 023737 001124 001126      CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
034402 001401      BEQ 76$ ;BR IF OK
034404 104007      EMT 7
034406 000240      76$: NOP
034410 000004      1$: SCOPE ;LOOP ?
  
```

783
802
803

```

*****
*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 RMD5.
*
* 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.
*****
  
```

```

034412 005737 001300      TST26: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
034412 001406      BEQ 2$ ;BR IF NOT
034420 100002      BPL 1$ ;BR IF JUST ENTERED TEST
034422 000137 003074      JMP EXEC ;RETURN & GET NEXT TEST NUMBER
034426 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
034434 012737 034450 001106 2$: MOV #TEST26,$LPADR ;SETUP SCOPE LOOP ADDRESS
034442 012737 034450 001110      MOV #TEST26,$LPERR ;SETUP ERROR LOOP ADDRESS
034450      TEST26:
  
```

804
837

```

034450 112737 000026 001102      MOVB  #26,$STSTNM      ;MOVE #26 TO TEST NUMBER
034456 012706 001100                MOV   #STACK,$SP      ;LOAD THE STACK POINTER
034462 012737 000012 001176      MOV   #10,$TIMES     ;;DO 10. ITERATIONS
  
```

;SET ATTENTION BITS FOR BOTH PORTS

```

034470 113760 001224 000010      MOVB  PORTA,$RMCS2(R0) ;SELECT PORT 64$
034476 005760 000012                TST   RMD5(R0)        ;MAKE SURE DRIVE AVAILABLE
034502 001775                BEQ   66$
034504 012760 177777 000014      MOV   #-1,$RMR1(R0)   ;FORCE ERRORS
034512 005060 000014                CLR   RMR1(R0)        ;CLEAR THE ERRORS
034516 013760 001226 000010      MOV   PORTB,$RMCS2(R0) ;SELECT THE OTHER PORT
034524 005760 000012                TST   RMD5(R0)        ;WAIT FOR DRIVE TO TIMEOUT
034530 001775                BEQ   64$             ;BR IF DRIVE HASN'T TIMED OUT
034532 012760 177777 000014      MOV   #-1,$RMR1(R0)   ;FORCE ERRORS ON PORT 65$
034540 005060 000014                CLR   RMR1(R0)        ;CLEAR THE ERRORS
034544 113760 001224 000010      MOVB  PORTA,$RMCS2(R0) ;SELECT PORT '64$' AGAIN
034552 005760 000012                TST   RMD5(R0)        ;WAIT FOR DRIVE TO TIMEOUT
034556 001775                BEQ   65$             ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

034560 113760 001224 000010      MOVB  PORTA,$RMCS2(R0) ;SELECT PORT A
034566 013737 001224 001240      MOV   PORTA,$PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034574 005037 001250                CLR   CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
034600 016037 000012 001126      MOV   RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
034606 012737 000012 001122      MOV   #RMD5,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034614 060037 001122                ADD   R0,$BDADR       ;ADD RH/RM BASE ADDRESS
034620 012737 100000 001124      MOV   #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
034626 013737 001126 001164      MOV   $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
034634 042737 077777 001164      BIC   #^CATA,$STMP0  ;SAVE SPECIFIED BITS
034642 023737 001124 001164      CMP   $GDDAT,$STMP0  ;COMPARE THE BITS
034650 001414                BEQ   67$             ;BR IF OK
034652 013737 001126 001174      MOV   $BDDAT,$STMP4   ;COPY 'BAD DATA'
034660 042737 100000 001174      BIC   #ATA,$STMP4    ;CLEAR THE MASKED BITS
034666 053737 001174 001124      BIS   $STMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
034674 104010                EMT   10
034676 005137 001250                COM   CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
034702 000240                NOP
034704 005737 001250                TST   CKERR           ;WAS ATTN BIT FOR PORT A SET ?
034710 001402                BEQ   +6              ;BR IF IT WAS
034712 000137 036104                JMP   1$              ;BYPASS REST OF TEST IF NOT
034716 113760 001226 000010      MOVB  PORTB,$RMCS2(R0) ;SELECT PORT B
034724 013737 001226 001240      MOV   PORTB,$PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
034732 005037 001250                CLR   CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
034736 016037 000012 001126      MOV   RMD5(R0),$BDDAT ;GET CONTENTS OF RMD5
034744 012737 000012 001122      MOV   #RMD5,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
034752 060037 001122                ADD   R0,$BDADR       ;ADD RH/RM BASE ADDRESS
034756 012737 100000 001124      MOV   #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
034764 013737 001126 001164      MOV   $BDDAT,$STMP0   ;MOVE REGISTER CONTENTS TO '$STMP0'
034772 042737 077777 001164      BIC   #^CATA,$STMP0  ;SAVE SPECIFIED BITS
035000 023737 001124 001164      CMP   $GDDAT,$STMP0  ;COMPARE THE BITS
035006 001414                BEQ   69$             ;BR IF OK
035010 013737 001126 001174      MOV   $BDDAT,$STMP4   ;COPY 'BAD DATA'
035016 042737 100000 001174      BIC   #ATA,$STMP4    ;CLEAR THE MASKED BITS
035024 053737 001174 001124      BIS   $STMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
  
```

```

035032 104010          EMT      10
035034 005137 001250  COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
035040 000240          NOP
035042 005737 001250  69$:   TST      CKERR      ;WAS ATTN BIT FOR PORT B SET ?
035046 001402          BEQ      +6        ;BR IF IT WAS
035050 000137 036104  JMP      1$        ;BYPASS REST OF TEST IF NOT
    
```

:SEIZE THE DRIVE THROUGH PORT A

```

035054 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
035062 013737 001224 001242  MOV    PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
035070 005060 000012          CLR    RMDS(R0) ;WRITE RMDS
035074 113760 001226 000010  MOVB   PORTB, RMCS2(R0) ;SELECT PORT B
035102 013737 001226 001240  MOV    PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035110 013737 001226 001244  MOV    PORTB, OPPRT ;'OPPOSITE' PORT ADDRESS
035116 016037 000012 001126  MOV    RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
035124 010037 001122          MOV    R0, $BDADR ;RH/RM BASE ADDRESS
035130 062737 000012 001122  ADD    #RMDS, $BDADR ;GENERATE BAD REGISTER ADDRESS
035136 005037 001124          CLR    $GDDAT ;REGISTER SHOULD BE ZERO
035142 023737 001124 001126  CMP    $GDDAT, $BDDAT ;IS THE REGISTER ZERO
035150 001403          BEQ    71$ ;BR IF IT IS
035152 104004          EMT      4
035154 000137 036104  JMP      1$ ;BYPASS REST OF THE SUBTEST
035160
    
```

71\$:

```

035160 113760 001224 000010  MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
035166 013737 001224 001240  MOV    PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035174 016037 000012 001126  MOV    RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
035202 042737 020001 001126  BIC    #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
035210 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
035216 013737 001124 001166  MOV    $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
035224 005137 001166          COM    $TMP1 ;COMPLEMENT THE EXPECTED STATUS
035230 013737 001126 001164  MOV    $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
035236 043737 001166 001164  BIC    $TMP1, $TMP0 ;CLEAR UNWANTED BITS
035244 023737 001124 001164  CMP    $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
035252 001401          BEQ    72$ ;BR IF THEY ARE
035254 104005          EMT      5
035256 000240          NOP
    
```

72\$:

:ISSUE DRIVE CLEAR COMMAND TO PORT A

```

035260 012760 000011 000000  MOV    #11, RMCS1(R0) ;DO A DRIVE CLEAR COMMAND
    
```

:VERIFY THAT ATTENTION BIT FOR PORT A CLEARED

```

035266 005037 001250          CLR    CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
035272 016037 000012 001126  MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
035300 012737 000012 001122  MOV    #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
035306 060037 001122          ADD    R0, $BDADR ;ADD RH/RM BASE ADDRESS
035312 005037 001124          CLR    $GDDAT ;WHAT REGISTER SHOULD BE
035316 013737 001126 001164  MOV    $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
035324 042737 077777 001164  BIC    #^CATA, $TMP0 ;SAVE SPECIFIED BITS
035332 023737 001124 001164  CMP    $GDDAT, $TMP0 ;COMPARE THE BITS
035340 001414          BEQ    73$ ;BR IF OK
035342 013737 001126 001174  MOV    $BDDAT, $TMP4 ;COPY 'BAD DATA'
035350 042737 100000 001174  BIC    #ATA, $TMP4 ;CLEAR THE MASKED BITS
035356 053737 001174 001124  BIS    $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
035364 104047          EMT      47
    
```

```

035366 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
035372 000240                73$:    NOP

;RELEASE THE DRIVE FROM PORT A

035374 113760 001224 000010    MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
035402 013737 001224 001240    MOV    PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035410 012760 000013 000000    MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

035416 005037 001254          CLR     RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
035422 012737 000012 001122    MOV    #RMDS, $BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
035430 060037 001122          ADD    R0, $BDADR    ;ADD THE I/O BASE ADDRESS
035434 012737 011700 001124    MOV    #MOL.PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
035442 113760 001224 000010    MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
035450 016037 000012 001170    MOV    RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
035456 042737 024001 001170    BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
035464 013737 001170 001164    MOV    $TMP2, $TMP0   ;COPY IT INTO '$TMP0'
035472 042737 100100 001164    BIC    #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035500 113760 001226 000010    MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
035506 016037 000012 001172    MOV    RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
035514 042737 024001 001172    BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
035522 013737 001172 001166    MOV    $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
035530 042737 100100 001166    BIC    #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
035536 023737 001164 001166    CMP    $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
035544 001006                BNE    75$           ;BR IF NOT
035546 005737 001164          TST    $TMP0         ;REGISTERS ARE THE SAME. ARE THEY ZERO ?
035552 001045                BNE    77$           ;BR IF NOT
035554 104046                EMT    46
035556 000137 035756          JMP    79$           ;BYPASS THE REST OF THE CHECKS
035562 013737 001170 001126 75$:  MOV    $TMP2, $BDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
035570 013737 001226 001240    MOV    PORTB, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035576 113760 001226 000010    MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
035604 005737 001164          TST    $TMP0         ;SEE IF STATUS EQ 0 FROM PORT A.
035610 001414                BEQ    76$           ;BR IF ZERO
035612 013737 001224 001240    MOV    PORTA, PTNBR   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
035620 013737 001172 001126    MOV    $TMP3, $BDAT   ;'BAD DATA' FOR ERROR TYPE CUT
035626 113760 001224 000010    MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
035634 005737 001166          TST    $TMP1         ;SEE IF STATUS EQ ZERO FROM PORT B.
035640 001012                BNE    77$           ;BR IF NOT
035642 012737 177777 001254 76$:  MOV    #-1, RELERR    ;SET 'RELEASE ERROR' INDICATOR
035650 012760 000011 000000    MOV    #11, RMCS1(R0) ;CLEAR THE DRIVE
035656 012760 000013 000000    MOV    #13, RMCS1(R0) ;RELEASE THE DRIVE
035664 104026                EMT    26
035666 013737 001170 001126 77$:  MOV    $TMP2, $BDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
035674 013737 001224 001240    MOV    PORTA, PTNBR   ;CHANGE PORT NUMBER
035702 042737 100000 001126    BIC    #ATA, $BDAT    ;DON'T CHECK THE ATTN BIT
035710 023737 001124 001126    CMP    $GDDAT, $BDAT ;ALL BITS OK ?
035716 001401                BEQ    78$           ;BR IF OK FROM PORT A.
035720 104007                EMT    7
035722 013737 001172 001126 78$:  MOV    $TMP3, $BDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
035730 013737 001226 001240    MOV    PORTB, PTNBR   ;CHANGE PORT NUMBER
035736 042737 100000 001126    BIC    #ATA, $BDAT    ;DON'T CHECK THE ATTN BIT
035744 023737 001124 001126    CMP    $GDDAT, $BDAT ;SEE IF READ OK FROM PORT B.
035752 001401                BEQ    79$           ;BR IF OK
035754 104007                EMT    7
  
```

```

035756 000240          79$:  NOP
                                ;CHECK ATTENTION BIT ON THE OPPOSITE PORT (PORT B)
035760 113760 001226 000010  MOVB  PORTB, RMCS?(R0) ;SELECT PORT B
035766 013737 001226 001240  MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
035774 005037 001250          CLR   CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
036000 016037 000012 001126  MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036006 012737 000012 001122  MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036014 060037 001122          ADD   R0, $BDADR    ;ADD RH/RM BASE ADDRESS
036020 012737 100000 001124  MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
036026 013737 001126 001164  MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
036034 042737 077777 001164  BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
036042 023737 001124 001164  CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
036050 001414          BEQ   80$          ;BR IF OK
036052 013737 001126 001174  MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
036060 042737 100000 001174  BIC   #ATA, $TMP4  ;CLEAR THE MASKED BITS
036066 053737 001174 001124  BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
036074 104050          EMT   50
036076 005137 001250          COM   CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
036102 000240          80$:  NOP
036104 000004          1$:   SCOPE
                                ;LOOP ?
    
```

838
856
857

```

:*****
:*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.
:*
:*****
    
```

```

036106          TST27:
036106 005737 001300  TST   KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
036112 001406          BEQ   2$          ;BR IF NOT
036114 100002          BPL   1$          ;BR IF JUST ENTERED TEST
036116 000137 003074  JMP   EXEC        ;RETURN & GET NEXT TEST NUMBER
036122 012737 177777 001300 1$:  MOV   #-1, KYBCTL ;SET SINGLE TEST INDICATOR
036130 012737 036144 001106 2$:  MOV   #TEST27, $LPADR ;SETUP SCOPE LOOP ADDRESS
036136 012737 036144 001110  MOV   #TEST27, $LPERR ;SETUP ERROR LOOP ADDRESS
036144          TEST27:
036144 112737 000027 001102  MOVB  #27, $STSTM ;MOVE #27 TO TEST NUMBER
036152 012706 001100          MOV   #STACK, SP  ;LOAD THE STACK POINTER
036156 012737 000012 001176  MOV   #10., $TIMES ;DO 10. ITERATIONS
    
```

858
859

;SET ATTENTION BITS FOR BOTH PORTS


```

036164 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT 64$
036172 005760 000012      TST   RMDS(R0)        ;MAKE SURE DRIVE AVAILABLE
036176 001775                BEQ   66$
036200 012760 177777 000014      MOV   #-1, RMER1(R0)  ;FORCE ERRORS
036206 005060 000014      CLR   RMER1(R0)      ;CLEAR THE ERRORS
036212 013760 001226 000010      MOV   PORTB, RMCS2(R0);SELECT THE OTHER PORT
036220 005760 000012      TST   RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
036224 001775                BEQ   64$
036226 012760 177777 000014      MOV   #-1, RMER1(R0)  ;FORCE ERRORS ON PORT 65$
036234 005060 000014      CLR   RMER1(R0)      ;CLEAR THE ERRORS
036240 113760 001224 000010      MOVB  PORTA, RMCS2(R0);SELECT PORT '64$' AGAIN
036246 005760 000012      TST   RMDS(R0)        ;WAIT FOR DRIVE TO TIMEOUT
036252 001775                BEQ   65$
    
```

:CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

036254 113760 001226 000010      MOVB  PORTB, RMCS2(R0);SELECT PORT B
036262 013737 001226 001240      MOV   PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036270 005037 001250                CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
036274 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036302 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036310 060037 001122                ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
036314 012737 100000 001124      MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
036322 013737 001126 001164      MOV   $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
036330 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
036336 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
036344 001414                BEQ   67$
036346 013737 001126 001174      MOV   $BDDAT, $TMP4  ;COPY 'BAD DATA'
036354 042737 100000 001174      BIC   #ATA, $TMP4   ;CLEAR THE MASKED BITS
036362 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
036370 104010                EMT   10
036372 005137 001250                COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
036376 000240                NOP
036400 005737 001250      67$:  TST   CKERR          ;WAS ATTN BIT FOR PORT B SET ?
036404 001402                BEQ   +6            ;BR IF IT WAS
036406 000137 037600                JMP   1$           ;BYPASS REST OF TEST IF NOT
036412 113760 001224 000010      MOVB  PORTA, RMCS2(R0);SELECT PORT A
036420 013737 001224 001240      MOV   PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036426 005037 001250                CLR   CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
036432 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
036440 012737 000012 001122      MOV   #RMDS, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
036446 060037 001122                ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
036452 012737 100000 001124      MOV   #ATA, $GDDAT  ;WHAT REGISTER SHOULD BE
036460 013737 001126 001164      MOV   $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
036466 042737 077777 001164      BIC   #^CATA, $TMP0 ;SAVE SPECIFIED BITS
036474 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
036502 001414                BEQ   69$
036504 013737 001126 001174      MOV   $BDDAT, $TMP4  ;COPY 'BAD DATA'
036512 042737 100000 001174      BIC   #ATA, $TMP4   ;CLEAR THE MASKED BITS
036520 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
036526 104010                EMT   10
036530 005137 001250                COM   CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
036534 000240                NOP
036536 005737 001250      69$:  TST   CKERR          ;WAS ATTN BIT FOR PORT A SET ?
036542 001402                BEQ   +6            ;BR IF IT WAS
036544 000137 037600                JMP   1$           ;BYPASS REST OF TEST IF NOT
    
```


:SEIZE THE DRIVE THROUGH PORT B

```

036550 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SFLECT PORT B
036556 013737 001226 001242      MOV   PORTB, SEIZPT ;STORE SEIZING PORT'S ADDRESS
036564 005060 000012      CLR   RMD5(R0) ;WRITE RMD5
036570 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
036576 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036604 013737 001224 001244      MOV   PORTA, OPRT ;'OPPOSITE' PORT ADDRESS
036612 016037 000012 001126      MOV   RMD5(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
036620 010037 001122      MOV   R0, $BDADR ;RH/RM BASE ADDRESS
036624 062737 000012 001122      ADD   #RMD5, $BDADR ;GENERATE BAD REGISTER ADDRESS
036632 005037 001124      CLR   $GDDAT ;REGISTER SHOULD BE ZERO
036636 023737 001124 001126      CMP   $GDDAT, $BDDAT ;IS THE REGISTER ZERO
036644 001403      BEQ   71$ ;BR IF IT IS
036646 104004      EMT   4
036650 000137 037600      JMP   1$ ;BYPASS REST OF THE SUBTEST
036654 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
036662 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
036670 016037 000012 001126      MOV   RMD5(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
036676 042737 020001 001126      BIC   #OM!PIP, $BDDAT ;CLEAR DONT CARE BITS
036704 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
036712 013737 001124 001166      MOV   $GDDAT, $TMP1 ;USE GOOD DATA AS A MASK
036720 005137 001166      COM   $TMP1 ;COMPLEMENT THE EXPECTED STATUS
036724 013737 001126 001164      MOV   $BDDAT, $TMP0 ;SAVE THE ACTUAL STATUS
036732 043737 001166 001164      BIC   $TMP1, $TMP0 ;CLEAR UNWANTED BITS
036740 023737 001124 001164      CMP   $GDDAT, $TMP0 ;ARE THE EXPECTED STATUS BITS SET ?
036746 001401      BEQ   72$ ;BR IF THEY ARE
036750 104005      EMT   5
036752 000240      NOP
    
```

:ISSUE DRIVE CLEAR COMMAND TO PORT B

```

036754 012760 000011 000000      MOV   #11, RMCS1(R0) ;DO A DRIVE CLEAR COMMAND
    
```

:VERIFY THAT ATTENTION BIT FOR PORT B CLEARED

```

036762 005037 001250      CLR   CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
036766 016037 000012 001126      MOV   RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
036774 012737 000012 001122      MOV   #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037002 050037 001122      ADD   R0, $BDADR ;ADD RH/RM BASE ADDRESS
037006 005037 001124      CLR   $GDDAT ;WHAT REGISTER SHOULD BE
037012 013737 001126 001164      MOV   $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
037020 042737 077777 001164      BIC   #*CATA, $TMP0 ;SAVE SPECIFIED BITS
037026 023737 001124 001164      CMP   $GDDAT, $TMP0 ;COMPARE THE BITS
037034 001414      BEQ   73$ ;BR IF OK
037036 013737 001126 001174      MOV   $BDDAT, $TMP4 ;COPY 'BAD DATA'
037044 042737 100000 001174      BIC   #ATA, $TMP4 ;CLEAR THE MASKED BITS
037052 053737 001174 001124      BIS   $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
037060 104047      EMT   47
037062 005137 001250      COM   CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
037066 000240      NOP
    
```

:RELEASE THE DRIVE FROM PORT B

```

037070 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
    
```

```

037076 013737 001226 001240      MOV   PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037104 012760 000013 000000      MOV   #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

037112 005037 001254      CLR   RELERR      ;CLEAR THE 'RELEASE ERROR ' INDICATOR
037116 012737 000012 001122      MOV   #RMDS,$BADDR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
037124 060037 001122      ADD   RO,$BADDR   ;ADD THE I/O BASE ADDRESS
037130 012737 011700 001124      MOV   #MCL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
037136 113760 001224 000010      MOV   PORTA,RMCS2(RO) ;SELECT PORT A.
037144 016037 000012 001170      MOV   RMDS(RO),STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
037152 042737 024001 001170      BIC   #PIP!WRL!OM,STMP2 ;CLEAR DONT CARES
037160 013737 001170 001164      MOV   STMP2,STMP0 ;COPY IT INTO 'STMP0'
037166 042737 100100 001164      BIC   #ATA!VV,STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037174 113760 001226 000010      MOV   PORTB,RMCS2(RO) ;SELECT PORT B.
037202 016037 000012 001172      MOV   RMDS(RO),STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
037210 042737 024001 001172      BIC   #PIP!WRL!OM,STMP3 ;CLEAR DONT CARES
037216 013737 001172 001166      MOV   STMP3,STMP1 ;COPY IT INTO 'STMP1'
037224 042737 100100 001166      BIC   #ATA!VV,STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
037232 023737 001164 001166      CMP   STMP0,STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
037240 001006      BNE   75$        ;BR IF NOT
037242 005737 001164      TST   STMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
037246 001045      BNE   77$        ;BR IF NOT
037250 104046      EMT   46
037252 000137 037452      JMP   79$        ;BYPASS THE REST OF THE CHECKS
037256 013737 001170 001126 75$:  MOV   STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
037264 013737 001226 001240      MOV   PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037272 113760 001226 000010      MOV   PORTB,RMCS2(RO) ;SELECT PORT B.
037300 005737 001164      TST   STMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
037304 001414      BEQ   76$        ;BR IF ZERO
037306 013737 001224 001240      MOV   PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
037314 013737 001172 001126      MOV   STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
037322 113760 001224 000010      MOV   PORTA,RMCS2(RO) ;SELECT PORT A.
037330 005737 001166      TST   STMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
037334 001012      BNE   77$        ;BR IF NOT
037336 012737 177777 001254 76$:  MOV   #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
037344 012760 000011 000000      MOV   #11,RMCS1(RO) ;CLEAR THE DRIVE
037352 012760 000013 000000      MOV   #13,RMCS1(RO) ;RELEASE THE DRIVE
037360 104026      EMT   26
037362 013737 001170 001126 77$:  MOV   STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
037370 013737 001224 001240      MOV   PORTA,PTNBR ;CHANGE PORT NUMBER
037376 042737 100000 001126      BIC   #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
037404 023737 001124 001126      CMP   $GDDAT,$BDDAT ;ALL BITS OK ?
037412 001401      BEQ   78$        ;BR IF OK FROM PORT A.
037414 104007      EMT   7
037416 013737 001172 001126 78$:  MOV   STMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
037424 013737 001226 001240      MOV   PORTB,PTNBR ;CHANGE PORT NUMBER
037432 042737 100000 001126      BIC   #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
037440 023737 001124 001126      CMP   $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
037446 001401      BEQ   79$        ;BR IF OK
037450 104007      EMT   7
037452 000240      NOP

```

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

```

037454 113760 001224 000010      MOV   PORTA,RMCS2(RO) ;SELECT PORT A
037462 013737 001224 001240      MOV   PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

```

037470 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
037474 016037 000012 001126 MOV RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
037502 012737 000012 001122 MOV #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
037510 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
037514 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
037522 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
037530 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
037536 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
037544 001414 BEQ 80$ ;BR IF OK
037546 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
037554 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
037562 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
037570 104050 EMT 50
037572 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
037576 000240 80$: NOP
037600 000004 1$: SCOPE ;LOOP ?

```

860
879
880

```

*****
*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 RMD5.
*
* 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.
*****

```

```

037602 TST30:
037602 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
037606 001406 BEQ 2$ ;BR IF NOT
037610 100002 BPL 1$ ;BR IF JUST ENTERED TEST
037612 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
037616 012737 177777 001300 1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
037624 012737 037640 001106 2$: MOV #TEST30, $LPADR ;SETUP SCOPE LOOP ADDRESS
037632 012737 037640 001110 MOV #TEST30, $LPERR ;SETUP ERROR LOOP ADDRESS
037640 TEST30:
037640 112737 000030 001102 MOVB #30, $STNUM ;MOVE #30 TO TEST NUMBER
037646 012706 001100 MOV #STACK, SP ;LOAD THE STACK POINTER
037652 012737 000012 001176 MOV #10, $TIMES ;DO 10. ITERATIONS

```

881
914

;SET ATTENTION BITS FOR BOTH PORTS

```

037660 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT 64$
037666 005760 000012 66$: TST RMD5(R0) ;MAKE SURE DRIVE AVAILABLE
037672 001775 BEQ 66$
037674 012760 177777 000014 MOV #-1, RMER1(R0) ;FORCE ERRORS

```

```

037702 005060 000014          CLR    RMER1(R0)      ;CLEAR THE ERRORS
037706 013760 001226 000010  MOV    PORTB,RMCS2(R0) ;SELECT THE OTHER PORT
037714 005760 000012          TST    RMD5(R0)      ;WAIT FOR DRIVE TO TIMEOUT
64$:   037720 001775          BEQ    64$           ;BR IF DRIVE HASN'T TIMED OUT
037722 012760 177777 000014  MOV    #-1,RMER1(R0)  ;FORCE ERRORS ON PORT 65$
037730 005060 000014          CLR    RMER1(R0)      ;CLEAR THE ERRORS
037734 013760 001224 000010  MOVB   PORTA,RMCS2(R0) ;SELECT PORT '64$' AGAIN
037742 005760 000012          TST    RMD5(R0)      ;WAIT FOR DRIVE TO TIMEOUT
037746 001775          BEQ    65$           ;BR IF DRIVE HASN'T TIMED OUT
  
```

.CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

037750 013760 001224 000010  MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
037756 013737 001224 001240  MOV    PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
037764 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
037770 016037 000012 001126  MOV    RMD5(R0),SBDDAT ;GET CONTENTS OF RMD5
037776 012737 000012 001122  MOV    #RMD5,SBADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040004 060037 001122          ADD    R0,SBADR      ;ADD RH/RM BASE ADDRESS
040010 012737 100000 001124  MOV    #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
040016 013737 001126 001164  MOV    SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
040024 042737 077777 001164  BIC    #^CATA,$TMP0   ;SAVE SPECIFIED BITS
040032 023737 001124 001164  CMP    $GDDAT,$TMP0   ;COMPARE THE BITS
040040 001414          BEQ    67$           ;BR IF OK
040042 013737 001126 001174  MOV    SBDDAT,$TMP4   ;COPY 'BAD DATA'
040050 042737 100000 001174  BIC    #ATA,$TMP4     ;CLEAR THE MASKED BITS
040056 053737 001174 001124  BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
040064 104010          EMT    10
040066 005137 001250          COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
040072 000240          NOP
040074 005737 001250          TST    CKERR          ;WAS ATTENTION SET FOR A??
040100 001402          BEQ    .+6           ;YES!!
040102 000137 041274          JMP    1$            ;NO - BYPASS REST OF TEST
040106 013760 001226 000010  MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
040114 013737 001226 001240  MOV    PORTB,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040122 005037 001250          CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
040126 016037 000012 001126  MOV    RMD5(R0),SBDDAT ;GET CONTENTS OF RMD5
040134 012737 000012 001122  MOV    #RMD5,SBADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040142 060037 001122          ADD    R0,SBADR      ;ADD RH/RM BASE ADDRESS
040146 012737 100000 001124  MOV    #ATA,$GDDAT    ;WHAT REGISTER SHOULD BE
040154 013737 001126 001164  MOV    SBDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
040162 042737 077777 001164  BIC    #^CATA,$TMP0   ;SAVE SPECIFIED BITS
040170 023737 001124 001164  CMP    $GDDAT,$TMP0   ;COMPARE THE BITS
040176 001414          BEQ    69$           ;BR IF OK
040200 013737 001126 001174  MOV    SBDDAT,$TMP4   ;COPY 'BAD DATA'
040206 042737 100000 001174  BIC    #ATA,$TMP4     ;CLEAR THE MASKED BITS
040214 053737 001174 001124  BIS    $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
040222 104010          EMT    10
040224 005137 001250          COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
040230 000240          NOP
040232 005737 001250          TST    CKERR          ;WAS ATTENTION SET FOR B??
040236 001402          BEQ    .+6           ;YES!!
040240 000137 041274          JMP    1$            ;NO - BYPASS REST OF TEST
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

040244 013760 001224 000010  MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
040252 013737 001224 001240  MOV    PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
  
```

```

040260 005060 000012          CLR      RMDS(R0)          ;WRITE RMDS
040264 113760 001226 000010  MOV#B   PORTB, RMCS2(R0)   ;SELECT PORT B
040272 013737 001226 001240  MOV     PORTB, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040300 013737 001226 001244  MOV     PORTB, OPPRT     ;'OPPOSITE' PORT ADDRESS
040306 016037 000012 001126  MOV     RMDS(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT A
040314 010037 001122          MOV     R0, $BDADR      ;RH/RM BASE ADDRESS
040320 062737 000012 001122  ADD     #RMDS, $BDADR   ;GENERATE BAD REGISTER ADDRESS
040326 005037 001124          CLR     $GDDAT         ;REGISTER SHOULD BE ZERO
040332 023737 001124 001126  CMP     $GDDAT, $BDDAT  ;IS THE REGISTER ZERO
040340 001403  BEQ     71$           ;BR IF IT IS
040342 104004  EMT     4
040344 000137 041274  JMP     1$           ;BYPASS REST OF THE SUBTEST
040350          71$:
040350 113760 001224 000010  MOV#B   PORTA, RMCS2(R0)   ;SELECT PORT A
040356 013737 001224 001240  MOV     PORTA, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040364 016037 000012 001126  MOV     RMDS(R0), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
040372 042737 020001 001126  BIC     #OM!PIP, $BDDAT  ;CLEAR DONT CARE BITS
040400 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV, $GDDAT ;EXPECTED STATUS
040406 013737 001124 001166  MOV     $GDDAT, $TMP1   ;USE GOOD DATA AS A MASK
040414 005137 001166          COM     $TMP1           ;COMPLEMENT THE EXPECTED STATUS
040420 013737 001126 001164  MOV     $BDDAT, $TMP0   ;SAVE THE ACTUAL STATUS
040426 043737 001166 001164  BIC     $TMP1, $TMP0    ;CLEAR UNWANTED BITS
040434 023737 001124 001164  CMP     $GDDAT, $TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
040442 001401  BEQ     72$           ;BR IF THEY ARE
040444 104005  EMT     5
040446 000240          72$:
                                NOP
                                ;ISSUE NOP COMMAND TO PORT A
040450 012760 000001 000000  MOV     #1, RMCS1(R0)
                                ;VERIFY THAT ATTENTION FOR PORT A CLEARED
040456 005037 001250          CLR     CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
040462 016037 000012 001126  MOV     RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
040470 012737 000012 001122  MOV     #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
040476 060037 001122          ADD     R0, $BDADR     ;ADD RH/RM BASE ADDRESS
040502 005037 001124          CLR     $GDDAT         ;WHAT REGISTER SHOULD BE
040506 013737 001126 001164  MOV     $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
040514 042737 077777 001164  BIC     #^CATA, $TMP0   ;SAVE SPECIFIED BITS
040522 023737 001124 001164  CMP     $GDDAT, $TMP0   ;COMPARE THE BITS
040530 001414  BEQ     73$           ;BR IF OK
040532 013737 001126 001174  MOV     $BDDAT, $TMP4   ;COPY 'BAD DATA'
040540 042737 100000 001174  BIC     #ATA, $TMP4     ;CLEAR THE MASKED BITS
040546 053737 001174 001124  BIS     $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
040554 104061  EMT     61
040556 005137 001250          COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
040562 000240          73$:
                                NOP
                                ;RELEASE THE DRIVE FROM PORT A
040564 113760 001224 000010  MOV#B   PORTA, RMCS2(R0)   ;SELECT PORT A
040572 013737 001224 001240  MOV     PORTA, PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
040600 012760 000013 000000  MOV     #13, RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A
                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

```

040606 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
040612 012737 C00012 001122 MUV #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
040620 060037 001122 ADD RO,$BDADR ;ADD THE I/O BASE ADDRESS
040624 012737 011700 001124 MOV #MOL!PGM.DPR!DRY.VV,$GDDAT ;COMPARISON CONSTANT
040632 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A.
040640 016037 000012 001170 MOV RMDS(RO),$STMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
040646 042737 024001 001170 BIC #PIP!WRL.OM,$STMP2 ;CLEAR DONT CARES
040654 013737 001170 001164 MOV $STMP2,$STMP0 ;COPY IT INTO '$STMP0'
040662 042737 100100 001164 BIC #ATA!VV,$STMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040670 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B.
040676 016037 000012 001172 MOV RMDS(RO),$STMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
040704 042737 024001 001172 BIC #PIP!WRL.OM,$STMP3 ;CLEAR DONT CARES
040712 013737 001172 001166 MOV $STMP3,$STMP1 ;COPY IT INTO '$STMP1'
040720 042737 100100 001166 BIC #ATA.VV,$STMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
040726 023737 001164 001166 CMP $STMP0,$STMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
040734 001006 BNE 75$ ;BR IF NOT
040736 005737 001164 TST $STMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
040742 001045 BNE 77$ ;BR IF NOT
040744 104046 EMT 46
040746 000137 041146 JMP 79$ ;BYPASS THE REST OF THE CHECKS
040752 013737 001170 001126 75$: MOV $STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
040760 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
040766 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B.
040774 005737 001164 TST $STMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
041000 001414 BEQ 76$ ;BR IF ZERO
041002 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
041010 013737 001172 001126 MOV $STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
041016 113760 001224 000010 MOV# PORTA,RMCS2(RO) ;SELECT PORT A.
041024 005737 001166 TST $STMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
041030 001012 BNE 77$ ;BR IF NOT
041032 012737 177777 001254 76$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
041040 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
041046 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
041054 104026 EMT 26
041056 013737 001170 001126 77$: MOV $STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
041064 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
041072 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
041100 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
041106 001401 BEQ 78$ ;BR IF OK FROM PORT A.
041110 104007 EMT 7
041112 013737 001172 001126 78$: MOV $STMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
041120 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
041126 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
041134 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
041142 001401 BEQ 79$ ;BR IF OK
041144 104007 EMT 7
041146 000240 79$: NOP

```

;VERIFY THAT ATTENTION FOR PORT B IS STIL SET

```

041150 113760 001226 000010 MOV# PORTB,RMCS2(RO) ;SELECT PORT B
041156 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041164 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
041170 016037 000012 001126 MOV RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
041176 012737 000012 001122 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041204 060037 001122 ADD RO,$BDADR ;ADD RH/RM BASE ADDRESS
041210 012737 100000 001124 MOV #A,$GDDAT ;WHAT REGISTER SHOULD BE

```

```

041216 013737 001126 001164      MOV      $BDDAT,$STMP0      ;MOVE REGISTER CONTENTS TO '$STMP0'
041224 042737 077777 001164      BIC      #^CATA,$STMP0     ;SAVE SPECIFIED BITS
041232 023737 001124 001164      CMP      $GDDAT,$STMP0     ;COMPARE THE BITS
041240 001414                      BEQ      80$                ;BR IF OK
041242 013737 001126 001174      MOV      $BDDAT,$STMP4     ;COPY 'BAD DATA'
041250 042737 100000 001174      BIC      #ATA,$STMP4       ;CLEAR THE MASKED BITS
041256 053737 001174 001124      BIS      $STMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
041264 104062                      EMT
041266 005137 001250                      COM      CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
041272 000240                      80$: NOP
041274 000004                      1$: SCOPE
  
```

915
934
935

```

*****
*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.
*****
  
```

```

041276 005737 001300      TST      KYBCTL             ;PERFORMING ONLY SINGLE TEST ?
041276 001406                      BEQ      2$                 ;BR IF NOT
041302 100002                      BPL      1$                 ;BR IF JUST ENTERED TEST
041306 000137 003074      JMP      EXEC               ;RETURN & GET NEXT TEST NUMBER
041312 012737 177777 001300 1$: MOV      #-1,KYBCTL         ;SET SINGLE TEST INDICATOR
041320 012737 041334 001106 2$: MOV      #TEST31,$LPADR   ;SETUP SCOPE LOOP ADDRESS
041326 012737 041334 001110      MOV      #TEST31,$LPERR    ;SETUP ERROR LOOP ADDRESS
041334                      TEST31:
041334 112737 000031 001102      MOV      #31,$STSTM        ;MOVE #31 TO TEST NUMBER
041342 012706 001100                      MOV      #STACK,$SP        ;LOAD THE STACK POINTER
041346 012737 000012 001176      MOV      #10,$TIMES        ;DO 10. ITERATIONS
  
```

936
937

;SET ATTENTION BITS FOR BOTH PORTS

```

041354 113760 001224 000010      MOV      PORTA,RMCS2(R0)   ;SELECT PORT 64$
041362 005760 000012 66$: TST      RMDS(R0)         ;MAKE SURE DRIVE AVAILABLE
041366 001775                      BEQ      66$
041370 012760 177777 000014      MOV      #-1,RMER1(R0)    ;FORCE ERRORS
041376 005060 000014                      CLR      RMER1(R0)         ;CLEAR THE ERRORS
041402 013760 001226 000010      MOV      PORTB,RMCS2(R0)   ;SELECT THE OTHER PORT
041410 005760 000012 64$: TST      RMDS(R0)         ;WAIT FOR DRIVE TO TIMEOUT
041414 001775                      BEQ      64$                ;BR IF DRIVE HASN'T TIMED OUT
041416 012760 177777 000014      MOV      #-1,RMER1(R0)    ;FORCE ERRORS ON PORT 65$
  
```



```

041424 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
041430 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
041436 005760 000012 65$: TST RMD5(R0) ;WAIT FOR DRIVE TO TIMEOUT
041442 001775 BEQ 65$ ;BR IF DRIVE HASN'T TIMED OUT
  
```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

041444 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
041452 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041460 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
041464 016037 000012 001126 MOV RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
041472 012737 000012 001122 MOV #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041500 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
041504 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
041512 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
041520 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
041526 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
041534 001414 BEQ 67$ ;BR IF OK
041536 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
041544 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
041552 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
041560 104010 EMT 10
041562 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
041566 000240 67$: NOP
041570 005737 001250 TST CKERR ;WAS ATTENTION SET FOR B??
041574 001402 BEQ .+6 ;YES!!
041576 000137 042770 JMP 1$ ;NO - BYPASS REST OF TEST
041602 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
041610 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041616 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
041622 016037 000012 001126 MOV RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
041630 012737 000012 001122 MOV #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
041636 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
041642 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
041650 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
041656 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
041664 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
041672 001414 BEQ 69$ ;BR IF OK
041674 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
041702 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
041710 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
041716 104010 EMT 10
041720 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
041724 000240 69$: NOP
041726 005737 001250 TST CKERR ;WAS ATTENTION SET FOR A??
041732 001402 BEQ .+6 ;YES!!
041734 000137 042770 JMP 1$ ;NO - BYPASS REST OF TEST
  
```

;SEIZE THE DRIVE THROUGH PORT B

```

041740 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
041746 013737 001226 001240 MOV PORTB, SEIZPT ;STOR! SEIZING PORT'S ADDRESS
041754 005060 000012 CLR RMD5(R0) ;WRITE RMD5
041760 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A
041766 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
041774 013737 001224 001244 MOV PORTA, OPPRT ;'OPPOSITE' PORT ADDRESS
042002 016037 000012 001126 MOV RMD5(R0), $BDDAT ;SEE IF DRIVE SEIZED BY PORT B
  
```



```

042010 010037 001122      MOV      RO,$BDADR      ;RH/RM BASE ADDRESS      ;
042014 062737 000012 001122  ADD      #RMDS,$BDADR   ;GENERATE BAD REGISTER ADDRESS
042022 005037 001124      CLR      $GDDAT        ;REGISTER SHOULD BE ZERO
042026 023737 001124 001126  CMP      $GDDAT,$BDDAT  ;IS THE REGISTER ZERO
042034 001403      BEQ     71$           ;BR IF IT IS
042036 104004      EMT     4
042040 000137 042770      JMP     1$           ;BYPASS REST OF THE SUBTEST
042044      71$:
042044 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B
042052 013737 001226 001240  MOV     PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042060 016037 000012 001126  MOV     RMDS(RO), $BDDAT ;SEE IF SEIZING PORT SEES CORRECT STATUS
042066 042737 020001 001126  BIC     #OM!PIP,$BDDAT  ;CLEAR DONT CARE BITS
042074 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;EXPECTED STATUS
042102 013737 001124 001166  MOV     $GDDAT,$TMP1   ;USE GOOD DATA AS A MASK
042110 005137 001166      COM     $TMP1          ;COMPLEMENT THE EXPECTED STATUS
042114 013737 001126 001164  MOV     $BDDAT,$TMP0   ;SAVE THE ACTUAL STATUS
042122 043737 001166 001164  BIC     $TMP1,$TMP0    ;CLEAR UNWANTED BITS
042130 023737 001124 001164  CMP     $GDDAT,$TMP0   ;ARE THE EXPECTED STATUS BITS SET ?
042136 001401      BEQ     72$           ;BR IF THEY ARE
042140 104005      EMT     5
042142 000240      72$:  NOP

;ISSUE NOP COMMAND TO PORT B

042144 012760 000001 000000      MOV     #1, RMCS1(RO)

;VERIFY THAT ATTENTION FOR PORT B CLEARED

042152 005037 001250      CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
042156 016037 000012 001126  MOV     RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
042164 012737 000012 001122  MOV     #RMDS,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042172 060037 001122      ADD     RO,$BDADR      ;ADD RH/RM BASE ADDRESS
042176 005037 001124      CLR     $GDDAT        ;WHAT REGISTER SHOULD BE
042202 013737 001126 001164  MOV     $BDDAT,$TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
042210 042737 077777 001164  BIC     #^CATA,$TMP0   ;SAVE SPECIFIED BITS
042216 023737 001124 001164  CMP     $GDDAT,$TMP0   ;COMPARE THE BITS
042224 001414      BEQ     73$           ;BR IF OK
042226 013737 001126 001174  MOV     $BDDAT,$TMP4   ;COPY 'BAD DATA'
042234 042737 100000 001174  BIC     #ATA,$TMP4     ;CLEAR THE MASKED BITS
042242 053737 001174 001124  BIS     $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
042250 104061      EMT     61
042252 005137 001250      COM     CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
042256 000240      73$:  NOP

;RELEASE THE DRIVE FROM PORT B

042260 113760 001226 000010  MOVB    PORTB, RMCS2(RO) ;SELECT PORT B
042266 013737 001226 001240  MOV     PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042274 012760 000013 000000  MOV     #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

042302 005037 001254      CLR     RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
042306 012737 000012 001122  MOV     #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOU'
042314 060037 001122      ADD     RO,$BDADR      ;ADD THE I/O BASE ADDRESS
042320 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
042326 113760 001224 000010  MOVB    PORTA, RMCS2(RO) ;SELECT PORT A.
  
```

```

042334 016037 000012 001170 MOV RMDS(RO), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
042342 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
042350 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
042356 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
042364 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
042372 016037 000012 001172 MOV RMDS(RO), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
042400 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
042406 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
042414 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
042422 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
042430 001006 BNE 75$ ;BR IF NOT
042432 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
042436 001045 BNE 77$ ;BR IF NOT
042440 104046 EMT 46
042442 000137 042642 JMP 79$ ;BYPASS THE REST OF THE CHECKS
042446 013737 001170 001126 75$: MOV $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
042454 013737 001226 001240 MOV PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
042462 113760 001226 000010 MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
042470 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
042474 001414 BEQ 76$ ;BR IF ZERO
042476 013737 001224 001240 MOV PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
042504 013737 001172 001126 MOV $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
042512 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
042520 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
042524 001012 BNE 77$ ;BR IF NOT
042526 012737 177777 001254 76$: MOV #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
042534 012760 000011 000000 MOV #11, RMCS1(RO) ;CLEAR THE DRIVE
042542 012760 000013 000000 MOV #13, RMCS1(RO) ;RELEASE THE DRIVE
042550 104026 EMT 26
042552 013737 001170 001126 77$: MOV $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
042560 013737 001224 001240 MOV PORTA, PTNBR ;CHANGE PORT NUMBER
042566 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
042574 023737 001124 001126 CMP $GDDAT, $BDDAT ;ALL BITS OK ?
042602 001401 BEQ 78$ ;BR IF OK FROM PORT A.
042604 104007 EMT 7
042606 013737 001172 001126 78$: MOV $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
042614 013737 001226 001240 MOV PORTB, PTNBR ;CHANGE PORT NUMBER
042622 042737 100000 001126 BIC #ATA, $BDDAT ;DON'T CHECK THE ATTN BIT
042630 023737 001124 001126 CMP $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
042636 001401 BEQ 79$ ;BR IF OK
042640 104007 EMT 7
042642 000240 79$: NOP

```

;VERIFY THAT ATTENTION FOR PORT A IS STIL SET

```

042644 113760 001224 000010 MOVB PORTA, RMCS2(RO) ;SELECT PORT A
042652 013737 001224 001240 MOV PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
042660 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
042664 016037 000012 001126 MOV RMDS(RO), $BDDAT ;GET CONTENTS OF RMDS
042672 012737 000012 001122 MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
042700 060037 001122 ADD RO, $BDADR ;ADD RH/RM BASE ADDRESS
042704 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
042712 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
042720 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
042726 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
042734 001414 BEQ 80$ ;BR IF OK
042736 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'

```

```

042744 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
042752 053737 001174 001124      BIS      $TMP4,$GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
042760 104062                                     EMT      62
042762 005137 001250                                     COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
042766 000240      80$:      NOP
042770 000004      1$:      SCOPE

```

938
952
953

```

:*****
:*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.
:*
:*****

```

```

042772
042772 005737 001300
042776 001406
043000 100002
043002 000137 003074
043006 012737 177777 001300
043014 012737 043030 001106
043022 012737 043030 001110
043030
043030 112737 000032 001102
043036 012706 001100
043042 012737 000012 001176

```

```

TST32:
      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      #TEST32,$LPADR ;SETUP SCOPE LOOP ADDRESS
      MOV      #TEST32,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST32:
      MOVB    #32,$TSTNM   ;MOVE #32 TO TEST NUMBER
      MOV     #STACK,SP    ;LOAD THE STACK POINTER
      MOV     #10.,$TIMES  ;DO 10. ITERATIONS

```

954
989

;SET ATTENTION BITS FOR BOTH PORTS

```

043050 113760 001224 000010      MOVB    PORTA, RMCS2(R0) ;SELECT PORT 64$
043056 005760 000012      66$:   TST      RMD5(R0)      ;MAKE SURE DRIVE AVAILABLE
043062 001775                                     BEQ      66$
043064 012760 177777 000014      MOV     #-1, RMER1(R0)   ;FORCE ERRORS
043072 005060 000014      CLR     RMER1(R0)       ;CLEAR THE ERRORS
043076 013760 00226 000010      MOV     PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
043104 005760 000012      64$:   TST      RMD5(R0)      ;WAIT FOR DRIVE TO TIMEOUT
043110 001775                                     BEQ      64$           ;BR IF DRIVE HASN'T TIMED OUT
043112 012760 177777 000014      MOV     #-1, RMER1(R0)   ;FORCE ERRORS ON PORT 65$
043120 005060 000014      CLR     RMER1(R0)       ;CLEAR THE ERRORS
043124 113760 001224 000010      MOVB    PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
043132 005760 000012      65$:   TST      RMD5(R0)      ;WAIT FOR DRIVE TO TIMEOUT
043136 001775                                     BEQ      65$           ;BR IF DRIVE HASN'T TIMED OUT

```

;CONFIRM THAT BOTH ATTENTION BITS ARE SET

```

043140 113760 001224 000010      MOVB    PORTA, RMCS2(R0) ;SELECT PORT A
043146 013737 001224 001240      MOV     PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043154 005037 001250      CLR     CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR

```

```

043160 016037 000012 001126 MOV RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
043166 012737 000012 001122 MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043174 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
043200 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
043206 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
043214 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
043222 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
043230 001414 BEQ 67$ ;BR IF OK
043232 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
043240 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
043246 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
043254 104010 EMT 10
043256 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
043262 000240 67$: NOP
043264 005737 001250 TST CKERR ;WAS ATTN BIT FOR PORT A SET ?
043270 001402 BEQ .+6 ;BR IF IT WAS
043272 000137 044246 JMP 1$ ;BYPASS REST OF TEST IF NOT
043276 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
043304 013737 001226 001240 MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
043312 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
043316 016037 000012 001126 MOV RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
043324 012737 000012 001122 MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
043332 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
043336 012737 100000 001124 MOV #ATA, $GDDAT ;WHAT REGISTER SHOULD BE
043344 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
043352 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
043360 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
043366 001414 BEQ 69$ ;BR IF OK
043370 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
043376 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
043404 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
043412 104010 EMT 10
043414 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
043420 000240 69$: NOP
043422 005737 001250 TST CKERR ;WAS ATTN BIT FOR PORT B SET ?
043426 001402 BEQ .+6 ;BR IF IT WAS
043430 000137 044246 JMP 1$ ;BYPASS REST OF TEST IF NOT

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

043434 005037 001254 CLR RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
043440 012737 000012 001122 MOV #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
043446 060037 001122 ADD R0, $BDADR ;ADD THE I/O BASE ADDRESS
043452 012737 111700 001124 MOV #111700, $GDDAT ;COMPARISON CONSTANT
043460 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT A.
043466 016037 000012 001170 MOV RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
043474 042737 024001 001170 BIC #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
043502 013737 001170 001164 MOV $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
043510 042737 100100 001164 BIC #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043516 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B.
043524 016037 000012 001172 MOV RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
043532 042737 024001 001172 BIC #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
043540 013737 001172 001166 MOV $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
043546 042737 100100 001166 BIC #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
043554 023737 001164 001166 CMP $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
043562 001006 BNE 71$ ;BR IF NOT
043564 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
  
```

```

043570 001045      BNE      73$      ;BR IF NOT
043572 104046      EMT      46
043574 000137 043760      JMP      75$      ;BYPASS THE REST OF THE CHECKS
043600 013737 001170 001126 71$:      MOV      STMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
043606 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043614 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B.
043622 005737 001164      TST      STMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
043626 001414      BEQ      72$      ;BR IF ZERO
043630 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
043636 013737 001172 001126      MOV      STMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
043644 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A.
043652 005737 001166      TST      STMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
043656 001012      BNE      73$      ;BR IF NOT
043660 012737 177777 001254 72$:      MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
043666 012760 000011 000000      MOV      #11,RMCS1(R0) ;CLEAR THE DRIVE
043674 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
043702 104026      EMT      26
043704 013737 001170 001126 73$:      MOV      STMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
043712 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
043720 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
043726 001401      BEQ      74$      ;BR IF OK FROM PORT A.
043730 104007      EMT      7
043732 013737 001172 001126 74$:      MOV      STMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
043740 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
043746 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
043754 001401      BEQ      75$      ;BR IF OK
043756 104007      EMT      7
043760 000240      NOP
043762 005737 001254 75$:      TST      RELERR      ;WAS DRIVE IN NEUTRAL ?
043766 001402      BEQ      +6      ;BR IF IT WAS
043770 000137 044246      JMP      1$      ;BYPASS RESET OF TEST
    
```

;ISSUE THE MASSBUS INIT

```

043774 012760 000040 000010      MOV      #CLR,RMCS2(R0) ;ISSUE A MASSBUS INIT
    
```

;CHECK THE ATTENTION BITS OF BOTH PORTS

```

044002 113760 001224 000010      MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
044010 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044016 005037 001250      CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
044022 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
044030 012737 000012 001122      MOV      #RMDS,$BADDR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044036 060037 001122      ADD      R0,$BADDR ;ADD RH/RM BASE ADDRESS
044042 005037 001124      CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
044046 013737 001126 001164      MOV      $BDDAT,$STMP0 ;MOVE REGISTER CONTENTS TO 'STMP0'
044054 042737 077777 001164      BIC      #^CATA,$STMP0 ;SAVE SPECIFIED BITS
044062 023737 001124 001164      CMP      $GDDAT,$STMP0 ;COMPARE THE BITS
044070 001414      BEQ      76$      ;BR IF OK
044072 013737 001126 001174      MOV      $BDDAT,$STMP4 ;COPY 'BAD DATA'
044100 042737 100000 001174      BIC      #ATA,$STMP4 ;CLEAR THE MASKED BITS
044106 053737 001174 001124      BIS      $STMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
044114 104051      EMT      51
044116 005137 001250      COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
044122 000240      NOP
044124 113760 001226 000010 76$:      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
044132 013737 001226 001240      MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

```

044140 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
044144 016037 000012 001126 MOV RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
044152 012737 000012 001122 MOV #RMD5, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044160 060037 001122 ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
044164 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
044170 013737 001126 001164 MOV $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
044176 042737 077777 001164 BIC #^CATA, $TMP0 ;SAVE SPECIFIED BITS
044204 023737 001124 001164 CMP $GDDAT, $TMP0 ;COMPARE THE BITS
044212 001414 BEQ 78$ ;BR IF OK
044214 013737 001126 001174 MOV $BDDAT, $TMP4 ;COPY 'BAD DATA'
044222 042737 100000 001174 BIC #ATA, $TMP4 ;CLEAR THE MASKED BITS
044230 053737 001174 001124 BIS $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
044236 104051 EMT 51
044240 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERPOR INDICATOR
044244 000240 78$: NOP
044246 000004 1$: SCOPE ;LOOP ?
    
```

993
1007
1008

```

*****
*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.
*****
    
```

```

044250 005737 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
044254 001406 BEQ 2$ ;BR IF NOT
044256 100002 BPL 1$ ;BR IF JUST ENTERED TEST
044260 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
044264 012737 177777 001300 1$: MOV #-1, KYBCTL ;SET SINGLE TEST INDICATOR
044272 012737 044306 001106 2$: MOV #TEST33, $LPADR ;SETUP SCOPE LOOP ADDRESS
044300 012737 044306 001110 MOV #TEST33, $LPERR ;SETUP ERROR LOOP ADDRESS
044306 TEST33:
044306 112737 000033 001102 MOVB #33, $STNM ;MOVE #33 TO TEST NUMBER
044314 012706 001100 MOV #STACK, SP ;LOAD THE STACK POINTER
044320 012737 000002 001176 MOV #2., $TIMES ;DO 2. ITERATIONS
    
```

1009
1053

;SET ATTENTION BITS FOR BOTH PORTS

```

044326 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT 64$
044334 005760 000012 66$: TST RMD5(R0) ;MAKE SURE DRIVE AVAILABLE
044340 001775 BEQ 66$
044342 012760 177777 000014 MOV #-1, RMER1(R0) ;FORCE ERRORS
044350 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
044354 013760 001226 000010 MOV PORTB, RMCS2(R0) ;SELECT THE OTHER PORT
044362 005760 000012 64$: TST RMD5(R0) ;WAIT FOR DRIVE TO TIMEOUT
044366 001775 BEQ 64$ ;BR IF DRIVE HASN'T TIMED OUT
044370 012760 177777 000014 MOV #-1, RMER1(R0) ;FORCE ERRORS ON PORT 65$
    
```

```

044376 005060 000014          CLR    RMER1(R0)      ;CLEAR THE ERRORS
044402 113760 001224 000010  MOVB  PORTA, RMCS2(R0) ;SELECT PORT '64$' AGAIN
044410 005760 000012          TST   RMDS(R0)       ;WAIT FOR DRIVE TO TIMEOUT
044414 001775          BEQ   65$           ;BR IF DRIVE HASN'T TIMED OUT

;CONFIRM THAT BOTH ATTENTION BITS ARE SET
044416 113760 001224 000010  MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
044424 013737 001224 001240  MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044432 005037 001250          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
044436 016037 000012 001126  MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
044444 012737 000012 001122  MOV   #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044452 060037 001122          ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
044456 012737 100000 001124  MOV   #ATA, $GDDAT   ;WHAT REGISTER SHOULD BE
044464 013737 001126 001164  MOV   $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
044472 042737 077777 001164  BIC   #^CATA, $TMP0  ;SAVE SPECIFIED BITS
044500 023737 001124 001164  CMP   $GDDAT, $TMP0  ;COMPARE THE BITS
044506 001414          BEQ   67$           ;BR IF OK
044510 013737 001126 001174  MOV   $BDDAT, $TMP4  ;COPY 'BAD DATA'
044516 042737 100000 001174  BIC   #ATA, $TMP4    ;CLEAR THE MASKED BITS
044524 053737 001174 001124  BIS   $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
044532 104010          EMT   10
044534 005137 001250          COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044540 000240          NOP
044542 005737 001250          TST   CKERR         ;WAS ATA SET FOR A??
044546 001402          BEQ   +6           ;YES - CONTINUE
044550 000137 045372          JMP   1$           ;BYPASS REST OF TEST
044554 113760 001226 000010  MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
044562 013737 001226 001240  MOV   PORTB, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
044570 005037 001250          CLR    CKERR         ;CLEAR THE 'CHECK ERROR' INDICATOR
044574 016037 000012 001126  MOV   RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
044602 012737 000012 001122  MOV   #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
044610 060037 001122          ADD   R0, $BDADR     ;ADD RH/RM BASE ADDRESS
044614 012737 100000 001124  MOV   #ATA, $GDDAT   ;WHAT REGISTER SHOULD BE
044622 013737 001126 001164  MOV   $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
044630 042737 077777 001164  BIC   #^CATA, $TMP0  ;SAVE SPECIFIED BITS
044636 023737 001124 001164  CMP   $GDDAT, $TMP0  ;COMPARE THE BITS
044644 001414          BEQ   69$           ;BR IF OK
044646 013737 001126 001174  MOV   $BDDAT, $TMP4  ;COPY 'BAD DATA'
044654 042737 100000 001174  BIC   #ATA, $TMP4    ;CLEAR THE MASKED BITS
044662 053737 001174 001124  BIS   $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
044670 104010          EMT   10
044672 005137 001250          COM   CKERR         ;SET THE REGISTER COMPARE ERROR INDICATOR
044676 000240          NOP
044700 005737 001250          TST   CKERR         ;WAS ATA SET FOR B??
044704 001402          BEQ   +6           ;YES - CONTINUE
044706 000137 045372          JMP   1$           ;BYPASS REST OF TEST

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
044712 005037 001254          CLR    RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
044716 012737 000012 001122  MOV   #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
044724 060037 001122          ADD   R0, $BDADR     ;ADD THE I/O BASE ADDRESS
044730 012737 111700 001124  MOV   #111700, $GDDAT ;COMPARISON CONSTANT
044736 113760 001224 000010  MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
044744 016037 000012 001170  MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
044752 042737 024001 001170  BIC   #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
044760 013737 001170 001164  MOV   $TMP2, $TMP0   ;COPY IT INTO '$TMP0'
  
```



```

044766 042737 100100 001164 BIC #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
044774 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
045002 016037 000012 001172 MOV RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
045010 042737 024001 001172 BIC #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
045016 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
045024 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
045032 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
045040 001006 BNE 71$ ;BR IF NOT
045042 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
045046 001045 BNE 73$ ;BR IF NOT
045050 104046 EMT 46
045052 000137 045236 JMP 75$ ;BYPASS THE REST OF THE CHECKS
045056 013737 001170 001126 71$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
045064 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045072 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
045100 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
045104 001414 BEQ 72$ ;BR IF ZERO
045106 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
045114 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
045122 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
045130 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
045134 001012 BNE 73$ ;BR IF NOT
045136 012737 177777 001254 72$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
045144 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
045152 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
045160 104026 EMT 26
045162 013737 001170 001126 73$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
045170 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
045176 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
045204 001401 BEQ 74$ ;BR IF OK FROM PORT A.
045206 104007 EMT 7
045210 013737 001172 001126 74$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
045216 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
045224 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
045232 001401 BEQ 75$ ;BR IF OK
045234 104007 EMT 7
045236 000240 75$: NOP
045240 005737 001254 TST RELERR ;WAS DRIVE IN NEUTRAL??
045244 001402 BEQ +6 ;YES!!
045246 000137 045372 JMP 1$ ;BYPASS REST OF TEST

```

;WRITE THE ATTENTION BIT

```

045252 013760 001236 000016 MOV ASR1,RMAS(R0)

```

;VERIFY THAT BOTH ATTENTIONS ARE RESET BY READING RMAS

```

045260 016037 000016 001126 MOV RMAS(R0),$BDDAT ;GET ATTENTION SUMMARY
045266 033737 001236 001126 BIT ASR1,$BDDAT ;IS THE ATTENTION RESET ??
045274 001414 BEQ 2$ ;YES !!
045276 010037 001122 MOV R0,$BDADR ;SETUP REGISTER ADDRESS
045302 062737 000016 001122 ADD #RMAS,$BDADR
045310 013737 001126 001124 MOV $BDDAT,$GDDAT ;SETUP EXPECTED DATA
045316 043737 001236 001124 BIC ASR1,$GDDAT ;RESET THIS DRIVES BIT
045324 104060 EMT 60

```

045326

2\$:

;WAIT FOR THE DRIVES TO RELEASE BY TIMEOUT


```

045326 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
045334 013737 001224 001240      MOV   PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045342 005760 000012                3$:  TST   RMDS(R0) ;MAKE SURE DEVICE IS AVAILABLE
045346 001775                        BEQ   3$
045350 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
045356 013737 001226 001240      MOV   PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045364 005760 000012                4$:  TST   RMDS(R0) ;MAKE SURE DEVICE IS AVAILABLE
045370 001775                        BEQ   4$
045372 000004                1$:  SCOPE
  
```

1054
1067
1068

```

*****
*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.
*****
  
```

```

045374 005737 001300      TST34:  TST   KYBCTL ;PERFORMING ONLY SINGLE TEST ?
045374 001406                BEQ   2$ ;BR IF NOT
045402 100002                BPL   1$ ;BR IF JUST ENTERED TEST
045404 000137 003074      JMP   EXEC ;RETURN & GET NEXT TEST NUMBER
045410 012737 177777 001300  1$:  MOV   #-1, KYBCTL ;SET SINGLE TEST INDICATOR
045416 012737 045432 001106  2$:  MOV   #TEST34, $LPADR ;SETUP SCOPE LOOP ADDRESS
045424 012737 045432 001110      MCV   #TEST34, $LPERR ;SETUP ERROR LOOP ADDRESS
045432                                TEST34:
045432 112737 000034 001102      MOVB  #34, $TSTNM ;MOVE #34 TO TEST NUMBER
045440 012706 001100      MOV   #STACK, SP ;LOAD THE STACK POINTER
045444 012737 000012 001176      MOV   #10., $TIMFS ;DO 10. ITERATIONS
  
```

1069
1103

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

045452 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
045460 005060 000012      CLR   RMDS(R0) ;SEIZE THE DRIVE
045464 012760 000011 000000      MOV   #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
045472 012760 000013 000000      MOV   #13, RMCS1(R0) ;RELEASE THE DRIVE
045500 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
045506 005060 000012      CLR   RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
045512 012760 000011 000000      MOV   #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
045520 012760 000013 000000      MOV   #13, RMCS1(R0) ;RELEASE THE DRIVE
045526 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
045534 012760 177777 000014      MCV   #-1, RMER1(R0) ;SET ERRORS TO FORCE ATTN BIT ON PORT A
045542 005060 000014      CLR   RMER1(R0) ;CLEAR THE ERRORS
045546 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
045554 005760 000012                1$:  TST   RMDS(R0) ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
045560 001775                        BEQ   1$ ;BR IF STILL SEIZED BY PORT A
045562 012737 000016 001122      MOV   #RMAS, $BDADR ;FORM ADDRESS OF ATTN REG IF ERROR
045570 060037 001122      ADD   R0, $BDADR ;ADD THE ADDRESS BASE
  
```

```

045574 013737 001236 001124      MOV      ASR1,$GDDAT      ;GOOD DATA FOR ERROR MESSAGE
045602 013737 001236 001166      MOV      ASR1,$TMP1      ;MAKE DATA COMPARE MASK
045610 005137 001166          COM      $TMP1          ;COMPLEMENT IT
045614 012737 045650 001110      MOV      #2$,$LPERR      ;LOAD LOOP ON ERROR ADDRESS
045622 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
045630 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045636 013737 001226 001242      MOV      PORTB,SEIZPT    ;'SEIZED' PORT ADDRESS
045644 005060 000012          CLR      RMDS(RO)       ;SEIZE THE DRIVE THROUGH PORT B
045650 016037 000016 001126 2$:  MOV      RMAS(RO),$BDDAT  ;GET THE CONTENTS OF THE ATTENTION REG
045656 013737 001126 001164      MOV      $BDDAT,$TMP0   ;PUT CONTENTS INTO WORKING LOCATION
045664 043737 001166 001164      BIC     $TMP1,$TMP0     ;CLEAR OTHER BITS
045672 023737 001124 001164      CMP     $GDDAT,$TMP0    ;SEE IF ATTN BIT FOR DRIVE SET
045700 001401          BEQ     3$              ;BR IF SET
045702 104053          EMT     3$              ;
045704          ;RELEASE THE DRIVE FROM PORT B

045704 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B
045712 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
045720 012760 000013 000000      MOV      #13,RMCS1(RO)  ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

045726 005037 001254          CLR      RELERR         ;CLEAR THE 'RELEASE ERROR' INDICATOR
045732 012737 000012 001122      MOV      #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
045740 060037 001122          ADD     RO,$BDADR       ;ADD THE I/O BASE ADDRESS
045744 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
045752 113760 001224 000010      MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
045760 016037 000012 001170      MOV      RMDS(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
045766 042737 024001 001170      BIC     #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
045774 013737 001170 001164      MOV      $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
046002 042737 100100 001164      BIC     #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046010 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B.
046016 016037 000012 001172      MOV      RMDS(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046024 042737 024001 001172      BIC     #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
046032 013737 001172 001166      MOV      $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
046040 042737 100100 001166      BIC     #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046046 023737 001164 001166      CMP     $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046054 001006          BNE     64$            ;BR IF NOT
046056 005737 001164          TST     $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046062 001045          BNE     66$            ;BR IF NOT
046064 104046          EMT     46             ;
046066 000137 046266          JMP     68$            ;BYPASS THE REST OF THE CHECKS
046072 013737 001170 001126 64$:  MOV      $TMP2,$BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
046100 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046106 113760 001226 000010      MOVVB   PORTB,RMCS2(RO)  ;SELECT PORT B.
046114 005737 001164          TST     $TMP0          ;SEE IF STATUS EQ 0 FROM PORT A.
046120 001414          BEQ     65$            ;BR IF ZERO
046122 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
046130 013737 001172 001126      MOV      $TMP3,$BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
046136 113760 001224 000010      MOVVB   PORTA,RMCS2(RO)  ;SELECT PORT A.
046144 005737 001166          TST     $TMP1          ;SEE IF STATUS EQ ZERO FROM PORT B.
046150 001012          BNE     66$            ;BR IF NOT
046152 012737 177777 001254 65$:  MOV      #-1,RELERR     ;SET 'RELEASE ERROR' INDICATOR
046160 012760 000011 000000      MOV      #11,RMCS1(RO)  ;CLEAR THE DRIVE
046166 012760 000013 000000      MOV      #13,RMCS1(RO)  ;RELEASE THE DRIVE
046174 104026          EMT     26             ;

```

```

046176 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
046204 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
046212 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
046220 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
046226 001401 BEQ 67$ ;BR IF OK FROM PORT A.
046230 104007 EMT 7
046232 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
046240 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
046246 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
046254 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
046262 001401 BEQ 68$ ;BR IF OK
046264 104007 EMT 7
046266 000240 68$: NOP
046270 000004 SCOPE ;LOOP ?
  
```

1104
 1117
 1118

```

*****
*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 RMD5.
*
* C. READ THE ATTENTION REGISTER & VERIFY THAT THE ATTENTION BIT
* FOR THE DRIVE IS SET.
*****
  
```

```

046272 005777 001300 TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
046272 001406 BEQ 2$ ;BR IF NOT
046300 100002 BPL 1$ ;BR IF JUST ENTERED TEST
046302 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
046306 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
046314 012737 046330 001106 2$: MOV #TEST35,$LPADR ;SETUP SCOPE LOOP ADDRESS
046322 012737 046330 001110 MOV #TEST35,$LPERR ;SETUP ERRGR LOOP ADDRESS
046330 TEST35:
046330 112737 000035 001102 MOVB #35,$TSTNM ;MOVE #35 TO TEST NUMBER
046336 012706 001100 MOV #STACK,SP ;LOAD THE STACK POINTER
046342 012737 000012 001176 MOV #10,$TIMES ;DO 10. ITERATIONS
  
```

1119
 1120

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

046350 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
046356 005060 000012 CLR RMD5(R0) ;SEIZE THE DRIVE
046362 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
046370 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
046376 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT #B
046404 005060 000012 CLR RMD5(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
046410 012760 000011 000000 MOV #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
046416 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
046424 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B
046432 012760 177777 000014 MOV #-1,RMER1(R0) ;SET ERRORS TO FORCE ATTN BIT ON PORT B
046440 005060 000014 CLR RMER1(R0) ;CLEAR THE ERRORS
046444 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
  
```

```

046452 005760 000012      1$:  TST  RMDS(RO)      ;WAIT FOR DRIVE TO RETURN TO NEUTRAL
046456 001775              BEQ  1$              ;BR IF STILL SEIZED BY PORT B
046460 012737 000016 001122  MOV  #RMAS,$BDADR   ;FORM ADDRESS OF ATTN REG IF ERROR
046466 060037 001122              ADD  RO,$BDADR      ;ADD THE ADDRESS BASE
046472 013737 001236 001124  MOV  ASR1,$GDDAT    ;GOOD DATA FOR ERROR MESSAGE
046500 013737 001236 001166  MOV  ASR1,$TMP1     ;MAKE DATA COMPARE MASK
046506 005137 001166              COM  $TMP1          ;COMPLEMENT IT
046512 012737 046546 001110  MOV  #2$,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
046520 113760 001224 000010  MOVB PORTA, RMCS2(RO) ;SELECT PORT A
046526 013737 001224 001240  MOV  PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046534 013737 001224 001242  MOV  PORTA,SEIZPT   ;'SEIZED' PORT ADDRESS
046542 005060 000012              CLR  RMDS(RO)       ;SEIZE THE DRIVE THROUGH PORT A
046546 016037 000016 001126  2$:  MOV  RMAS(RO),$BDDAT ;GET THE CONTENTS OF THE ATTENTION REG
046554 013737 001126 001164  MOV  $BDDAT,$TMP0   ;PUT CONTENTS INTO WORKING LOCATION
046562 043737 001166 001164  BIC  $TMP1,$TMP0    ;CLEAR OTHER BITS
046570 023737 001124 001164  CMP  $GDDAT,$TMP0   ;SEE IF ATTN BIT FOR DRIVE SET
046576 001401              BEQ  3$              ;BR IF SET
046600 104053              EMT  53
046602

3$:
;RELEASE THE DRIVE FROM PORT A

046602 113760 001224 000010  MOVB PORTA, RMCS2(RO) ;SELECT PORT A
046610 013737 001224 001240  MOV  PORTA,PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
046616 012760 000013 000000  MOV  #13, RMCS1(RO) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

046624 005037 001254              CLR  RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
046630 012737 000012 001122  MOV  #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
046636 060037 001122              ADD  RO,$BDADR      ;ADD THE I/O BASE ADDRESS
046642 012737 011700 001124  MOV  #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
046650 113760 001224 000010  MOVB PORTA, RMCS2(RO) ;SELECT PORT A.
046656 016037 000012 001170  MOV  RMDS(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
046664 042737 024001 001170  BIC  #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
046672 013737 001170 001164  MOV  $TMP2,$TMP0    ;COPY IT INTO '$TMP0'
046700 042737 100100 001164  BIC  #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046706 113760 001226 000010  MOVB PORTB, RMCS2(RO) ;SELECT PORT B.
046714 016037 000012 001172  MOV  RMDS(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
046722 042737 024001 001172  BIC  #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
046730 013737 001172 001166  MOV  $TMP3,$TMP1    ;COPY IT INTO '$TMP1'
046736 042737 100100 001166  BIC  #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
046744 023737 001164 001166  CMP  $TMP0,$TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
046752 001006              BNE  64$           ;BR IF NOT
046754 005737 001164              TST  $TMP0          ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
046760 001045              BNE  66$           ;BR IF NOT
046762 104046              EMT  46
046764 000137 047164              JMP  68$
046770 013737 001170 001126  64$: MOV  $TMP2,$BDDAT   ;BYPASS THE REST OF THE CHECKS
046776 013737 001226 001240  MOV  PORTB,PTNBR    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047004 113760 001226 000010  MOVB PORTB, RMCS2(RO) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047012 005737 001164              TST  $TMP0          ;SELECT PORT B.
047016 001414              BEQ  65$           ;SEE IF STATUS EQ 0 FROM PORT A.
047020 013737 001224 001240  MOV  PORTA,PTNBR    ;BR IF ZERO
047026 013737 001172 001126  MOV  $TMP3,$BDDAT   ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047034 113760 001224 000010  MOVB PORTA, RMCS2(RO) ;'BAD DATA' FOR ERROR TYPE OUT
047042 005737 001166              TST  $TMP1          ;SELECT PORT A.
047046 001012              BNE  66$           ;SEE IF STATUS EQ ZERO FROM PORT B.
;BR IF NOT
    
```

```

047050 012737 177777 001254 65$: MOV #1,RELERR ;SET 'RELEASE ERROR' INDICATOR
047056 012760 000011 000000 MOV #11,RMCS1(RO) ;CLEAR THE DRIVE
047064 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
047072 104026 EMT 26
047074 013737 001170 001126 66$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
047102 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
047110 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
047116 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
047124 001401 BEQ 67$ ;BR IF OK FROM PORT A.
047126 104007 EMT 7
047130 013737 001172 001126 67$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
047136 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
047144 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
047152 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
047160 001401 BEQ 68$ ;BR IF OK
047162 104007 EMT 7
047164 000240 68$: NOP
047166 000004 SCOPE ;LOOP ?
  
```

1121
1138
1139

```

*****
*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.
*****
  
```

```

047170 005737 001300 TST36: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
047174 001406 BEQ 2$ ;BR IF NOT
047176 100002 BPL 1$ ;BR IF JUST ENTERED TEST
047200 000137 003074 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
047204 012737 177777 001300 1$: MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
047212 012737 047226 001106 2$: MOV #TEST36,$LPADR ;SETUP SCOPE LOOP ADDRESS
047220 012737 047226 001110 MOV #TEST36,$LPERR ;SETUP ERROR LOOP ADDRESS
047226 TEST36: MOV #36,$STNM ;MOVE #36 TO TEST NUMBER
047226 112737 000036 001102 MOV #STACK,SP ;LOAD THE STACK POINTER
047234 012706 001100 MOV #10,$TIMES ;DO 10. ITERATIONS
047240 012737 000012 001176
  
```

1140
1169

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

047246 113760 001224 000010 MOV #PORTA,RMCS2(RO) ;SELECT PORT #A
047254 005060 000012 CLR RMDS(RO) ;SEIZF THE DRIVE
047260 012760 000011 000000 MOV #11,RMCS1(RO) ;ISSUE DRIVE CLEAR
047266 012760 000013 000000 MOV #13,RMCS1(RO) ;RELEASE THE DRIVE
  
```

```

047274 113760 001226 000010      MOV#B  PORTB, RMCS2(R0) ;SELECT PORT #B
047302 005060 000012 000000      CLR    RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
047306 012760 000011 000000      MOV    #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
047314 012760 000013 000000      MOV    #13, RMCS1(R0) ;RELEASE THE DRIVE
047322 113760 001224 000010      MOV#B  PORTA, RMCS2(R0) ;SELECT PORT A
047330 013737 001224 001240      MOV    PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047336 013737 001224 001242      MOV    PORTA, SEIZPT  ;'SEIZED' PORT ADDRESS

;DO A OFFSET THROUGH PORT A

047344 012760 000015 000000      MOV    #15, RMCS1(R0) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT A

;WAIT FOR DRIVE TO FINISH

047352 032760 000200 000012      BIT    #DRY, RMDS(R0) ;WAIT FOR DRIVE TO FINISH
047360 001774                      BEQ    .-6            ;BR IF NOT FINISHED

;CONFIRM THAT ATTENTION IS SET FOR PORT A

047362 005037 001250                      CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
047366 016037 000012 001126      MOV    RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
047374 012737 000012 001122      MOV    #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
047402 060037 001122                      ADD    R0, $BDADR      ;ADD RH/RM BASE ADDRESS
047406 012737 100000 001124      MOV    #ATA, $GDDAT    ;WHAT REGISTER SHOULD BE
047414 013737 001126 001164      MOV    $BDDAT, $TMP0   ;MOVE REGISTER CONTENTS TO '$TMP0'
047422 042737 077777 001164      BIC    #^CATA, $TMP0   ;SAVE SPECIFIED BITS
047430 023737 001124 001164      CMP    $GDDAT, $TMP0   ;COMPARE THE BITS
047436 001414                      BEQ    64$            ;BR IF OK
047440 013737 001126 001174      MOV    $BDDAT, $TMP4   ;COPY 'BAD DATA'
047446 042737 100000 001174      BIC    #ATA, $TMP4     ;CLEAR THE MASKED BITS
047454 053737 001174 001124      BIS    $TMP4, $GDDAT   ;'OR' WITH GOOD DATA FOR TYPEOUT
047462 104032                      EMT    32
047464 005137 001250                      COM    CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
047470 000240      64$: NOP

;RELEASE THE DRIVE FROM PORT A

047472 113760 001224 000010      MOV#B  PORTA, RMCS2(R0) ;SELECT PORT A
047500 013737 001224 001240      MOV    PORTA, PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
047506 012760 000013 000000      MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

047514 005037 001254                      CLR    RELERR         ;CLEAR THE 'RELEASE ERROR ' INDICATOR
047520 012737 000012 001122      MOV    #RMDS, $BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
047526 060037 001122                      ADD    R0, $BDADR      ;ADD THE I/O BASE ADDRESS
047532 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
047540 113760 001224 000010      MOV#B  PORTA, RMCS2(R0) ;SELECT PORT A.
047546 016037 000012 001170      MOV    RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
047554 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
047562 013737 001170 001164      MOV    $TMP2, $TMP0    ;COPY IT INTO '$TMP0'
047570 042737 100100 001164      BIC    #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
047576 113760 001226 000010      MOV#B  PORTB, RMCS2(R0) ;SELECT PORT B.
047604 016037 000012 001172      MOV    RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
047612 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
047620 013737 001172 001166      MOV    $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
047626 042737 100100 001166      BIC    #ATA.VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
  
```

```

047634 023737 001164 001166      CMP      $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS
047642 001006                      BNE      66$             ;BR IF NOT
047644 005737 001164              TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
047650 001045                      BNE      68$             ;BR IF NOT
047652 104046                      EMT      46
047654 000137 050040              JMP      70$             ;BYPASS THE REST OF THE CHECKS
047660 013737 001170 001126 66$:  MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
047666 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047674 113760 001226 000010      MOVVB   PORTB,RMCS2(R0)  ;SELECT PORT B.
047702 005737 001164              TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
047706 001414                      BEQ      67$             ;BR IF ZERO
047710 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
047716 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
047724 113760 001224 000010      MOVVB   PORTA,RMCS2(R0)  ;SELECT PORT A.
047732 005737 001166              TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
047736 001012                      BNE      68$             ;BR IF NOT
047740 012737 177777 001254 67$:  MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
047746 012760 000011 000000      MOV      #11,RMCS1(R0)   ;CLEAR THE DRIVE
047754 012760 000013 000000      MOV      #13,RMCS1(R0)   ;RELEASE THE DRIVE
047762 104026                      EMT      26
047764 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
047772 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
050000 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;ALL BITS OK ?
050006 001401                      BEQ      69$             ;BR IF OK FROM PORT A.
050010 104007                      EMT      7
050012 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT    ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
050020 013737 001226 001240      MOV      PORTB,PTNBR     ;CHANGE PORT NUMBER
050026 023737 001124 001126      CMP      $GDDAT,$BDDAT   ;SEE IF READ OK FROM PORT B.
050034 001401                      BEQ      70$             ;BR IF OK
050036 104007                      EMT      7
050040 000240                      NOP
050042 113760 001226 000010 70$:  MOVVB   PORTB,RMCS2(R0)  ;SELECT PORT B
050050 013737 001226 001240      MOV      PORTB,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT B

```

050056 005037 001250              CLR      CKERR           ;CLEAR THE 'CHECK ERROR' INDICATOR
050062 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
050070 012737 000012 001122      MOV      #RMDS,$BDADR    ;FORM REGISTER ADDRESS OF ERROR MESSAGE
050076 060037 001122              ADD      R0,$BDADR       ;ADD RH/RM BASE ADDRESS
050102 005037 001124              CLR      $GDDAT         ;WHAT REGISTER SHOULD BE
050106 013737 001126 001164      MOV      $BDDAT,$TMP0    ;MOVE REGISTER CONTENTS TO '$TMP0'
050114 042737 077777 001164      BIC      #^CATA,$TMP0    ;SAVE SPECIFIED BITS
050122 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
050130 001414                      BEQ      71$             ;BR IF OK
050132 013737 001126 001174      MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
050140 042737 100000 001174      BIC      #ATA,$TMP4      ;CLEAR THE MASKED BITS
050146 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
050154 104032                      EMT      32
050156 005137 001250              COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
050162 000240 71$:  NOP
    
```

050164 000004 SCOPE ;LOOP ?

1170
1186
1187

 ;*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.
: *
:*****
  
```

TST37:

```

050166
050166 005737 001300
050172 001406
050174 100002
050176 000137 003074
050202 012737 177777 001300
050210 012737 050224 001106
050216 012737 050224 001110
050224
050224 112737 000037 001102
050232 012706 001100
050236 012737 000012 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 #T37,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST37,$LPERR ;SETUP ERROR LOOP ADDRESS

TEST37:
MOVB #37,$TSTNM ;MOVE #37 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #10,$TIMES ;;DO 10. ITERATIONS
  
```

1188
1189

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

050244 113760 001224 000010
050252 005060 000012
050256 012760 000011 000000
050264 012760 000013 000000
050272 113760 001226 000010
050300 005060 000012
050304 012760 000011 000000
050312 012760 000013 000000
050320 113760 001226 000010
050326 013737 001226 001240
050334 013737 001226 001242
  
```

```

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
MOVB PORTB, RMCS2(R0) ;SELECT PORT B
MOV PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
MOV PORTB, SEIZPT ;'SEIZED' PORT ADDRESS
  
```

;DO A OFFSET THROUGH PORT B

```

050342 012760 000015 000000
  
```

```

MOV #15, RMCS1(R0) ;ISSUE A OFFSET INSTRUCTION THROUGH PORT B
  
```

;WAIT FOR DRIVE TO FINISH

```

050350 032760 000200 000012
050356 001774
  
```

```

BIT #DRY, RMDS(R0) ;WAIT FOR DRIVE TO FINISH
BEQ -6 ;BR IF NOT FINISHED
  
```

;CONFIRM THAT ATTENTION IS SET FOR PORT B

```

050360 005037 001250
050364 016037 000012 001126
050372 012737 000012 001122
050400 060037 001122
  
```

```

CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
MOV #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
ADD R0, $BDADR ;ADD RH/RM BASE ADDRESS
  
```



```

050404 012737 100000 001124      MOV      #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
050412 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
050420 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
050426 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
050434 001414                      BEQ      64$ ;BR IF OK
050436 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
050444 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
050452 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
050460 104032                      EMT      32
050462 005137 001250                      COM      CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
050466 000240      64$: NOP
  
```

;RELEASE THE DRIVE FROM PORT B

```

050470 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B
050476 013737 001226 001240      MOV      PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
050504 012760 000013 000000      MOV      #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
  
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

050512 005037 001254                      CLR      RELERR ;CLEAR THE 'RELEASE ERROR ' INDICATOR
050516 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
050524 060037 001122                      ADD      R0,$BDADR ;ADD THE I/O BASE ADDRESS
050530 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY.VV,$GDDAT ;COMPARISON CONSTANT
050536 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
050544 016037 000012 001170      MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
050552 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
050560 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
050566 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050574 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
050602 016037 000012 001172      MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
050610 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
050616 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
050624 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
050632 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
050640 001006                      BNE      66$ ;BR IF NOT
050642 005737 001164                      TST      $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
050646 001045                      BNE      68$ ;BR IF NOT
050650 104046                      EMT      46
050652 000137 051036                      JMP      70$ ;BYPASS THE REST OF THE CHECKS
050656 013737 001170 001126      66$: MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
050664 013737 001226 001240      MOV      PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050672 113760 001226 000010      MOV      PORTB, RMCS2(R0) ;SELECT PORT B.
050700 005737 001164                      TST      $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
050704 001414                      BEQ      67$ ;BR IF ZERO
050706 013737 001224 001240      MOV      PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
050714 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
050722 113760 001224 000010      MOV      PORTA, RMCS2(R0) ;SELECT PORT A.
050730 005737 001166                      TST      $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
050734 001012                      BNE      68$ ;BR IF NOT
050736 012737 177777 001254      67$: MOV      #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
050744 012760 000011 000000      MOV      #11, RMCS1(R0) ;CLEAR THE DRIVE
050752 012760 000013 000000      MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
050760 104026                      EMT      26
050762 013737 001170 001126      68$: MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD< READ
050770 013737 001224 001240      MOV      PORTA, PTNBR ;CHANGE PORT NUMBER
050776 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
  
```

```

051004 001401      BEQ      69$      ;BR IF OK FROM PORT A.
051006 104007      EMT      7
051010 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
051016 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
051024 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM POR' B.
051032 001401      BEQ      70$      ;BR IF OK
051034 104007      EMT      7
051036 000240      NOP
051040 113760 001224 000010 70$:  MOVB     PORTA,RMCS2(RO) ;SELECT PORT A
051046 013737 001224 001240      MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

```

;CONFIRM THAT ATTENTION IS NOT SET FOR PORT A

```

051054 005037 001250      CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
051060 016037 000012 001126      MOV      RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
051066 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051074 060037 001122      ADD      RO,$BDADR ;ADD RH/RM BASE ADDRESS
051100 005037 001124      CLR      $GDDAT ;WHAT REGISTER SHOULD BE
051104 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051112 042737 077777 001164      BIC      #^CATA,$TMP0 ;SAVE SPECIFIED BITS
051120 023737 001124 001164      CMP      $GDDAT,$TMP0 ;COMPARE THE BITS
051126 001414      BEQ      71$      ;BR IF OK
051130 013737 001126 001174      MOV      $BDDAT,$TMP4 ;COPY 'BAD DATA'
051136 042737 100000 001174      BIC      #ATA,$TMP4 ;CLEAR THE MASKED BITS
051144 053737 001174 001124      BIS      $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051152 104032      EMT      32
051154 005137 001250      COM     CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
051160 000240      NOP

```

```

051162 000004      SCOPE ;LOOP ?

```

1193
1202
1203
1204

```

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

```

1205
1206
1215
1216
1217

```

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

```

1218
1219
1240
1241

```

*****
*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'.
  
```

```

*****
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     #10, $TIMES   ;DO 10. ITERATIONS
  
```

1242
1284

```

;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
  
```

```

051242 113760 001224 000010      MOVB    PORTA,RMCS2(R0) ;SELECT PORT A
051250 013737 001224 001242      MOV     PORTA,SEIZPT   ;STORE SEIZING PORT'S ADDRESS
051256 012760 000001 000024      MOV     #DMD,RMMR1(R0) ;WRITE DMD INTO RMMR1
051264 013737 001226 001244      MOV     PORTB,OPPRT   ;'OPPOSITE' PORT ADDRESS
051272 012760 001001 000024      MOV     #DMD!MUR,RMMR1(R0) ;SET UNIT READY
051300 012760 000000 000024      MOV     #0,RMMR1(R0)  ;RESET DIAGNOSTIC MODE
  
```

```

;VERIFY THAT ATA = 1,VV = 0 FOR PORT A
CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
MOV      #RMDS,$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 '$TMP0'
BIC      #^ATA,$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
  
```

```

051306 005037 001250
051312 016037 000012 001126
051320 012737 000012 001122
051326 060037 001122
051332 012737 100000 001124
051340 013737 001126 001164
051346 042737 077777 001164
051354 023737 001124 001164
051362 001414
051364 013737 001126 001174
051372 042737 100000 001174
051400 053737 001174 001124
051406 104064
  
```

```

051410 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051414 000240          NOP
051416 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051422 016037 000012 001126 66$:  MOV     RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
051430 012737 000012 001122  MOV     #RMD5, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051436 060037 001122          ADD     R0, $BDADR    ;ADD RH/RM BASE ADDRESS
051442 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
051446 013737 001126 001164  MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051454 042737 177677 001164  BIC     #^CVV, $TMP0  ;SAVE SPECIFIED BITS
051462 023737 001124 001164  CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
051470 001414          BEQ     68$        ;BR IF OK
051472 013737 001126 001174  MOV     $BDDAT, $TMP4 ;COPY 'BAD DATA'
051500 042737 000100 001174  BIC     #VV, $TMP4    ;CLEAR THE MASKED BITS
051506 053737 001174 001124  BIS     $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051514 104065          EMT     65
051516 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051522 000240          NOP
  
```

```

051524 012760 000011 000000 ;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A
051532 012760 000021 000000  MOV     #11, RMCS1(R0) ;DRIVE CLEAR
  MOV     #21, RMCS1(R0) ;READ IN PRESET
  
```

```

051540 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051544 016037 000012 001126  MOV     RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
051552 012737 000012 001122  MOV     #RMD5, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051560 060037 001122          ADD     R0, $BDADR    ;ADD RH/RM BASE ADDRESS
051564 012737 000100 001124  MOV     #VV, $GDDAT   ;WHAT REGISTER SHOULD BE
051572 013737 001126 001164  MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051600 042737 177677 001164  BIC     #^CVV, $TMP0  ;SAVE SPECIFIED BITS
051606 023737 001124 001164  CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
051614 001414          BEQ     70$        ;BR IF OK
051616 013737 001126 001174  MOV     $BDDAT, $TMP4 ;COPY 'BAD DATA'
051624 042737 000100 001174  BIC     #VV, $TMP4    ;CLEAR THE MASKED BITS
051632 053737 001174 001124  BIS     $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051640 104013          EMT     13
  
```

```

051642 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051646 000240          NOP
051650 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
051654 016037 000012 001126  MOV     RMD5(R0), $BDDAT ;GET CONTENTS OF RMD5
051662 012737 000012 001122  MOV     #RMD5, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
051670 060037 001122          ADD     R0, $BDADR    ;ADD RH/RM BASE ADDRESS
051674 012737 011700 001124  MOV     #MOL!PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
051702 013737 001126 001164  MOV     $BDDAT, $TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
051710 042737 024007 001164  BIC     #^C153770, $TMP0 ;SAVE SPECIFIED BITS
051716 023737 001124 001164  CMP     $GDDAT, $TMP0 ;COMPARE THE BITS
051724 001414          BEQ     72$        ;BR IF OK
051726 013737 001126 001174  MOV     $BDDAT, $TMP4 ;COPY 'BAD DATA'
051734 042737 153770 001174  BIC     #153770, $TMP4 ;CLEAR THE MASKED BITS
051742 053737 001174 001124  BIS     $TMP4, $GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
051750 104010          EMT     10
051752 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
051756 000240          NOP
  
```

```

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

;RELEASE THE DRIVE FROM PORT A

```
051760 113760 001224 000010      MOV# PORTA, RMCS2(R0) ;SELECT PORT A
051766 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
051774 012760 000013 000000      MOV  #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
052002 005037 001254          CLR  RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
052006 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
052014 060037 001122          ADD  R0, $BDADR ;ADD THE I/O BASE ADDRESS
052020 012737 011600 001124      MOV  #MOL!PGM.DPR.DRY, $GDDAT ;COMPARISON CONSTANT
052026 113760 001224 000010      MOV# PORTA, RMCS2(R0) ;SELECT PORT A.
052034 016037 000012 001170      MOV  RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
052042 042737 024001 001170      BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
052050 013737 001170 001164      MOV  $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
052056 042737 100100 001164      BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052064 113760 001226 000010      MOV# PORTB, RMCS2(R0) ;SELECT PORT B.
052072 016037 000012 001172      MOV  RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
052100 042737 024001 001172      BIC  #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
052106 013737 001172 001166      MOV  $TMP3, $TMP1 ;COPY IT INTO '$TMP1'
052114 042737 100100 001166      BIC  #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052122 023737 001164 001166      CMP  $TMP0, $TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052130 001006          BNE  74$ ;BR IF NOT
052132 005737 001164          TST  $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052136 001037          BNE  76$ ;BR IF NOT
052140 104046          EMT  46
052142 000137 052326          JMP  78$ ;BYPASS THE REST OF THE CHECKS
052146 013737 001170 001126 74$: MOV  $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052154 013737 001226 001240      MOV  PORTB, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052162 113760 001226 000010      MOV# PORTB, RMCS2(R0) ;SELECT PORT B.
052170 005737 001164          TST  $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
052174 001414          BEQ  75$ ;BR IF ZERO
052176 013737 001224 001240      MOV  PORTA, PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
052204 013737 001172 001126      MOV  $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
052212 113760 001224 000010      MOV# PORTA, RMCS2(R0) ;SELECT PORT A.
052220 005737 001166          TST  $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
052224 001004          BNE  76$ ;BR IF NOT
052226 012737 177777 001254 75$: MOV  #-1, RELERR ;SET 'RELEASE ERROR' INDICATOR
052234 104022          EMT  22
052236 013737 001170 001126 76$: MOV  $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
052244 013737 001224 001240      MOV  PORTA, PTNBR ;CHANGE PORT NUMBER
052252 042737 100100 001126      BIC  #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
052260 023737 001124 001126      CMP  $GDDAT, $BDDAT ;ALL BITS OK ?
052266 001401          BEQ  77$ ;BR IF OK FROM PORT A.
052270 104007          EMT  7
052272 013737 001172 001126 77$: MOV  $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
052300 013737 001226 001240      MOV  PORTB, PTNBR ;CHANGE PORT NUMBER
052306 042737 100100 001126      BIC  #ATA!VV, $BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
052314 023737 001124 001126      CMP  $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
052322 001401          BEQ  78$ ;BR IF OK
052324 104007          EMT  7
052326 000240          NOP
052330 113760 001226 000010 78$: MOV# PORTB, RMCS2(R0) ;SELECT PORT B
052336 013737 001226 001240      MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052344 005037 001250          CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
052350 016037 000012 001126      MOV  RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
```

```

052356 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052364 060037 001122                ADD    RO,$BDADR   ;ADD RH/RM BASE ADDRESS
052370 012737 000000 001124      MOV    #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
052376 013737 001126 001164      MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
052404 042737 077777 001164      BIC    #^CATA,$TMP0 ;SAVE SPECIFIED BITS
052412 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
052420 001414                BEQ    79$         ;BR IF OK
052422 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
052430 042737 100000 001174      BIC    #ATA,$TMP4   ;CLEAR THE MASKED BITS
052436 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
052444 104064                EMT    64
052446 005137 001250                COM    CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
052452 000240                79$:  NOP
052454 005037 001250                CLR    CKERR       ;CLEAR THE 'CHECK ERROR' INDICATOR
052460 016037 000012 001126      MOV    RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
052466 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
052474 060037 001122                ADD    RO,$BDADR   ;ADD RH/RM BASE ADDRESS
052500 005037 001124                CLR    $GDDAT      ;WHAT REGISTER SHOULD BE
052504 013737 001126 001164      MOV    $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
052512 042737 177677 001164      BIC    #^CVV,$TMP0 ;SAVE SPECIFIED BITS
052520 023737 001124 001164      CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
052526 001414                BEQ    81$         ;BR IF OK
052530 013737 001126 001174      MOV    $BDDAT,$TMP4 ;COPY 'BAD DATA'
052536 042737 000100 001174      BIC    #VV,$TMP4   ;CLEAR THE MASKED BITS
052544 053737 001174 001124      BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
052552 104065                EMT    65
052554 005137 001250                COM    CKERR       ;SET THE REGISTER COMPARE ERROR INDICATOR
052560 000240                81$:  NOP

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B,
;THEN RELEASE PORT B
052562 012760 000011 000000      MOV    #11,RMCS1(RO) ;DRIVE CLEAR
052570 012760 000021 000000      MOV    #21,RMCS1(RO) ;READ IN PRESET
;RELEASE THE DRIVE FROM PORT B

052576 113760 001226 000010      MOV    PORTB,RMCS2(RO) ;SELECT PORT B
052604 013737 001226 001240      MOV    PORTB,PTNBR   ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
052612 012760 000013 000000      MOV    #13,RMCS1(RO) ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

052620 005037 001254                CLR    RELERR      ;CLEAR THE 'RELEASE ERROR' INDICATOR
052624 012737 000012 001122      MOV    #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
052632 060037 001122                ADD    RO,$BDADR   ;ADD THE I/O BASE ADDRESS
052636 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
052644 113760 001224 000010      MOV    PORTA,RMCS2(RO) ;SELECT PORT A.
052652 016037 000012 001170      MOV    RMDS(RO),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
052660 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
052666 013737 001170 001164      MOV    $TMP2,$TMP0  ;COPY IT INTO '$TMP0'
052674 042737 100100 001164      BIC    #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052702 113760 001226 000010      MOV    PORTB,RMCS2(RO) ;SELECT PORT B.
052710 016037 000012 001172      MOV    RMDS(RO),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
052716 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
052724 013737 001172 001166      MOV    $TMP3,$TMP1  ;COPY IT INTO '$TMP1'
052732 042737 100100 001166      BIC    #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
052740 023737 001164 001166      CMP    $TMP0,$TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
052746 001006                BNE    83$         ;BR IF NOT
  
```

052750	005737	001164		TST	\$TMP0		;REGISTERS ARE THE SAME: ARE THEY ZERO ?
052754	001037			BNE	85\$;BR IF NOT
052756	104046			EMT	46		
052760	000137	053144		JMP	87\$;BYPASS THE REST OF THE CHECKS
052764	013737	001170	001126	83\$:	MOV	\$TMP2,\$BDDAT	;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
052772	013737	001226	001240		MOV	PORTB,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053000	113760	001226	000010		MOVB	PORTB,RMCS2(R0)	;SELECT PORT B.
053006	005737	001164			TST	\$TMP0	;SEE IF STATUS EQ 0 FROM PORT A.
053012	001414				BEQ	84\$;BR IF ZERO
053014	013737	001224	001240		MOV	PORTA,PTNBR	;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
053022	013737	001172	001126		MOV	\$TMP3,\$BDDAT	; 'BAD DATA' FOR ERROR TYPE OUT
053030	113760	001224	000010		MOVB	PORTA,RMCS2(R0)	;SELECT PORT A.
053036	005737	001166			TST	\$TMP1	;SEE IF STATUS EQ ZERO FROM PORT B.
053042	001004				BNE	85\$;BR IF NOT
053044	012737	177777	001254	84\$:	MOV	#-1,RELERR	;SET 'RELEASE ERROR' INDICATOR
053052	104022				EMT	22	
053054	013737	001170	001126	85\$:	MOV	\$TMP2,\$BDDAT	;LOOK FOR BIT FAILURES WHEN RMD5 READ
053062	013737	001224	001240		MOV	PORTA,PTNBR	;CHANGE PORT NUMBER
053070	042737	100100	001126		BIC	#ATA.VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
053076	023737	001124	001126		CMP	\$GDDAT,\$BDDAT	;ALL BITS OK ?
053104	001401				BEQ	86\$;BR IF OK FROM PORT A.
053106	104007				EMT	7	
053110	013737	001172	001126	86\$:	MOV	\$TMP3,\$BDDAT	;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
053116	013737	001226	001240		MOV	PORTB,PTNBR	;CHANGE PORT NUMBER
053124	042737	100100	001126		BIC	#ATA!VV,\$BDDAT	;DON'T CHECK ATTN BIT OR VV BIT
053132	023737	001124	001126		CMP	\$GDDA1,\$BDDAT	;SEE IF READ OK FROM PORT B.
053140	001401				BEQ	87\$;BR IF OK
053142	104007				EMT	7	
053144	000240			87\$:	NOP		
053146	000004			50\$:	SCOPE		

1285
1306
1307

```

*****
;*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'.
*****
TST41:
TST      KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
BEQ      2$         ;BR IF NOT
  
```

053150
053150 005737 001300
053154 001406

1308
1309

```

053156 100002          BPL      1$          ;BR IF JUST ENTERED TEST
053160 000137 003074  JMP      EXEC        ;RETURN & GET NEXT TEST NUMBER
053164 012737 177777 001300 1$:      MOV      #-1,KYBCTL  ;SET SINGLE TEST INDICATOR
053172 012737 053206 001106 2$:      MOV      #TEST41,$LPADR ;SETUP SCOPE LOOP ADDRESS
053200 012737 053206 001110      MOV      #TEST41,$LPERR ;SETUP ERROR LOOP ADDRESS
053206          TEST41:
053206 112737 000041 001102      MOVB     #41,$TSTNM    ;MOVE #41 TO TEST NUMBER
053214 012706 001100      MOV      #STACK,SP   ;LOAD THE STACK POINTER
053220 012737 000012 001176      MOV      #10.,$TIMES ;DO 10. ITERATIONS
  
```

```

;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
  
```

```

053226 113760 001226 000010      MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
053234 013737 001226 001242      MOV      PORTB,SEIZPT  ;STORE SEIZING PORT'S ADDRESS
053242 012760 000001 000024      MOV      #DMD,RMMR1(R0) ;WRITE DMD INTO RMMR1
053250 013737 001224 001244      MOV      PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
053256 012760 001001 000024      MOV      #DMD!MUR,RMMR1(R0) ;SET UNIT READY
053264 012760 000000 000024      MOV      #0,RMMR1(R0)  ;RESET DIAGNOSTIC MODE
  
```

```

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

```

053272 005037 001250          CLR      CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
053276 016037 000012 001126      MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
053304 012737 000012 001122      MOV      #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053312 060037 001122          ADD      R0,$BDADR    ;ADD RH/RM BASE ADDRESS
053316 012737 100000 001124      MOV      #ATA,$GDDAT  ;WHAT REGISTER SHOULD BE
053324 013737 001126 001164      MOV      $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
053332 042737 077777 001164      BIC     #^CATA,$TMP0  ;SAVE SPECIFIED BITS
053340 023737 001124 001164      CMP     $GDDAT,$TMP0  ;COMPARE THE BITS
053346 001414          BEQ     66$          ;BR IF OK
053350 013737 001126 001174      MOV     $BDDAT,$TMP4  ;COPY 'BAD DATA'
053356 042737 100000 001174      BIC     #ATA,$TMP4   ;CLEAR THE MASKED BITS
053364 053737 001174 001124      BIS     $TMP4,$GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
053372 104064          EMT     64
053374 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
053400 000240          66$:      NOP
053402 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
053406 016037 000012 001126      MOV     RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
053414 012737 000012 001122      MOV     #RMDS,$BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053422 060037 001122          ADD     R0,$BDADR    ;ADD RH/RM BASE ADDRESS
053426 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
053432 013737 001126 001164      MOV     $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
053440 042737 177677 001164      BIC     #^CVV,$TMP0  ;SAVE SPECIFIED BITS
053446 023737 001124 001164      CMP     $GDDAT,$TMP0 ;COMPARE THE BITS
053454 001414          BEQ     68$          ;BR IF OK
053456 013737 001126 001174      MOV     $BDDAT,$TMP4 ;COPY 'BAD DATA'
053464 042737 000100 001174      BIC     #VV,$TMP4   ;CLEAR THE MASKED BITS
053472 053737 001174 001124      BIS     $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
053500 104065          EMT     65
053502 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
053506 000240          68$:      NOP
  
```

```

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT B
  
```



```

053510 012760 000011 000000      MOV    #11,RMCS1(R0)      ;DRIVE CLEAR
053516 012760 000021 000000      MOV    #21,RMCS1(R0)      ;READ IN PRESET

;VERIFY ATA = 0 AND VV = 1 FOR PORT B
053524 005037 001250      CLR    CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
053530 016037 000012 001126      MOV    RMDS(R0), $BDDAT   ;GET CONTENTS OF RMDS
053536 012737 000012 001122      MOV    #RMDS, $BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053544 060037 001122      ADD    R0, $BDADR        ;ADD RH/RM BASE ADDRESS
053550 012737 000100 001124      MOV    #VV, $GDDAT       ;WHAT REGISTER SHOULD BE
053556 013737 001126 001164      MOV    $BDDAT, $TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
053564 042737 177677 001164      BIC    #^CVV, $TMP0      ;SAVE SPECIFIED BITS
053572 023737 001124 001164      CMP    $GDDAT, $TMP0     ;COMPARE THE BITS
053600 001414      BEQ    70$              ;BR IF OK
053602 013737 001126 001174      MOV    $BDDAT, $TMP4     ;COPY 'BAD DATA'
053610 042737 000100 001174      BIC    #VV, $TMP4        ;CLEAR THE MASKED BITS
053616 053737 001174 001124      BIS    $TMP4, $GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
053624 104013      EMT    13
053626 005137 001250      COM    CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
053632 000240      NOP
053634 005037 001250      CLR    CKERR              ;CLEAR THE 'CHECK ERROR' INDICATOR
053640 016037 000012 001126      MOV    RMDS(R0), $BDDAT   ;GET CONTENTS OF RMDS
053646 012737 000012 001122      MOV    #RMDS, $BDADR     ;FORM REGISTER ADDRESS OF ERROR MESSAGE
053654 060037 001122      ADD    R0, $BDADR        ;ADD RH/RM BASE ADDRESS
053660 012737 011700 001124      MOV    #MOL.PGM!DPR!DRY!VV, $GDDAT ;WHAT REGISTER SHOULD BE
053666 013737 001126 001164      MOV    $BDDAT, $TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
053674 042737 024007 001164      BIC    #^C153770, $TMP0  ;SAVE SPECIFIED BITS
053702 023737 001124 001164      CMP    $GDDAT, $TMP0     ;COMPARE THE BITS
053710 001414      BEQ    72$              ;BR IF OK
053712 013737 001126 001174      MOV    $BDDAT, $TMP4     ;COPY 'BAD DATA'
053720 042737 153770 001174      BIC    #153770, $TMP4    ;CLEAR THE MASKED BITS
053726 053737 001174 001124      BIS    $TMP4, $GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
053734 104010      EMT    10
053736 005137 001250      COM    CKERR              ;SET THE REGISTER COMPARE ERROR INDICATOR
053742 000240      NOP

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

;RELEASE THE DRIVE FROM PORT B
053744 113760 001226 000010      MOV    PORTB, RMCS2(R0)   ;SELECT PORT B
053752 013737 001226 001240      MOV    PORTB, PTNBR      ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
053760 012760 000013 000000      MOV    #13, RMCS1(R0)    ;ISSUE RELEASE THROUGH PORT B

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL
053766 005037 001254      CLR    RELERR            ;CLEAR THE 'RELEASE ERROR' INDICATOR
053772 012737 000012 001122      MOV    #RMDS, $BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
054000 060037 001122      ADD    R0, $BDADR        ;ADD THE I/O BASE ADDRESS
054004 012737 011600 001124      MOV    #MOL!PGM!DPR!DRY, $GDDAT ;COMPARISON CONSTANT
054012 113760 001224 000010      MOV    PORTA, RMCS2(R0)  ;SELECT PORT A.
054020 016037 000012 001170      MOV    RMDS(R0), $TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054026 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
054034 013737 001170 001164      MOV    $TMP2, $TMP0     ;COPY IT INTO '$TMP0'
054042 042737 100100 001164      BIC    #ATA!VV, $TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054050 113760 001226 000010      MOV    PORTB, RMCS2(R0)  ;SELECT PORT B.
054056 016037 000012 001172      MOV    RMDS(R0), $TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
  
```

```

054064 042737 024001 001172 BIC #PIP!WRL.OM,$TMP3 ;CLEAR DONT CARES
054072 013737 001172 001166 MOV $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
054100 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054106 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054114 001006 BNE 74$ ;BR IF NOT
054116 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054122 001037 BNE 76$ ;BR IF NOT
054124 104046 EMT 46
054126 000137 054312 JMP 78$ ;BYPASS THE REST OF THE CHECKS
054132 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054140 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054146 113760 001226 000010 MOVB PORTB,RMCS2(R0) ;SELECT PORT B.
054154 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
054160 001414 BEQ 75$ ;BR IF ZERO
054162 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054170 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
054176 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A.
054204 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
054210 001004 BNE 76$ ;BR IF NOT
054212 012737 177777 001254 75$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
054220 104022 EMT 22
054222 013737 001170 001126 76$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDs READ
054230 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
054236 042737 100100 001126 BIC #ATA.VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
054244 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
054252 001401 BEQ 77$ ;BR IF OK FROM PORT A.
054254 104007 EMT 7
054256 013737 001172 001126 77$: MOV $TMP3,$BDDAT ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
054264 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
054272 042737 100100 001126 BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
054300 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
054306 001401 BEQ 78$ ;BR IF OK
054310 104007 EMT 7
054312 000240 78$: NOP
054314 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT A
054322 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054330 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
054334 016037 000012 001126 MOV RMDs(R0),$BDDAT ;GET CONTENTS OF RMDs
054342 012737 000012 001122 MOV #RMDs,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054350 060037 001122 ADD R0,$BADR ;ADD RH/RM BASE ADDRESS
054354 012737 100000 001124 MOV #ATA,$GDDAT ;WHAT REGISTER SHOULD BE
054362 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
054370 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
054376 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
054404 001414 BEQ 79$ ;BR IF OK
054406 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
054414 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
054422 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
054430 104064 EMT 64
054432 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
054436 000240 79$: NOP
054440 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
054444 016037 000012 001126 MOV RMDs(R0),$BDDAT ;GET CONTENTS OF RMDs
054452 012737 000012 001122 MOV #RMDs,$BADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
054460 060037 001122 ADD R0,$BADR ;ADD RH/RM BASE ADDRESS
054464 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
054470 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
  
```

```

054476 042737 177677 001164      BIC      #^CVV,$TMP0      ;SAVE SPECIFIED BITS
054504 023737 001124 001164      CMP      $GDDAT,$TMP0    ;COMPARE THE BITS
054512 001414                      BEQ      81$             ;BR IF OK
054514 013737 001126 001174      MOV      $BDDAT,$TMP4    ;COPY 'BAD DATA'
054522 042737 000100 001174      BIC      #VV,$TMP4       ;CLEAR THE MASKED BITS
054530 053737 001174 001124      BIS      $TMP4,$GDDAT    ;'OR' WITH GOOD DATA FOR TYPEOUT
054536 104065                      EMT      65
054540 005137 001250                      COM      CKERR           ;SET THE REGISTER COMPARE ERROR INDICATOR
054544 000240                      81$:      NOP

;ISSUE A DRIVE CLEAR AND A READ IN PRESET COMMAND TO PORT A,
;THEN RELEASE PORT A
054546 012760 000011 000000      MOV      #11,RMCS1(R0)   ;DRIVE CLEAR
054554 012760 000021 000000      MOV      #21,RMCS1(R0)   ;READ IN PRESET
;RELEASE THE DRIVE FROM PORT A

054562 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A
054570 013737 001224 001240      MOV      PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
054576 012760 000013 000000      MOV      #13,RMCS1(R0)   ;ISSUE RELEASE THROUGH PORT A

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

054604 005037 001254                      CLR      RELERR          ;CLEAR THE 'RELEASE ERROR ' INDICATOR
054610 012737 000012 001122      MOV      #RMDS,$BDADR    ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
054616 060037 001122                      ADD      R0,$BDADR       ;ADD THE I/O BASE ADDRESS
054622 012737 011600 001124      MOV      #MOL!PGM!DPR!DRY,$GDDAT ;COMPARISON CONSTANT
054630 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
054636 016037 000012 001170      MOV      RMDS(R0),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
054644 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
054652 013737 001170 001164      MOV      $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
054660 042737 100100 001164      BIC      #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054666 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
054674 016037 000012 001172      MOV      RMDS(R0),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
054702 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
054710 013737 001172 001166      MOV      $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
054716 042737 100100 001166      BIC      #ATA!VV,$TMP1   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
054724 023737 001164 001166      CMP      $TMP0,$TMP1     ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
054732 001006                      BNE      83$            ;BR IF NOT
054734 005737 001164                      TST      $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
054740 001037                      BNE      85$            ;BR IF NOT
054742 104046                      EMT      46
054744 000137 055130                      JMP      87$            ;BYPASS THE REST OF THE CHECKS
054750 013737 001170 001126 83$:      MOV      $TMP2,$BDDAT    ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
054756 013737 001226 001240      MOV      PORTB,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
054764 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT B.
054772 005737 001164                      TST      $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
054776 001414                      BEQ      84$            ;BR IF ZERO
055000 013737 001224 001240      MOV      PORTA,PTNBR     ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
055006 013737 001172 001126      MOV      $TMP3,$BDDAT    ;'BAD DATA' FOR ERROR TYPE OUT
055014 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT A.
055022 005737 001166                      TST      $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
055026 001004                      BNE      85$            ;BR IF NOT
055030 012737 177777 001254 84$:      MOV      #-1,RELERR      ;SET 'RELEASE ERROR' INDICATOR
055036 104022                      EMT      22
055040 013737 001170 001126 85$:      MOV      $TMP2,$BDDAT    ;LOOK FOR BIT FAILURES WHEN RMDS READ
055046 013737 001224 001240      MOV      PORTA,PTNBR     ;CHANGE PORT NUMBER
055054 042737 100100 001126      BIC      #ATA.VV,$BDDAT  ;DON'T CHECK ATTN BIT OR VV BIT
    
```

```

055062 023737 001124 001126
055070 001401
055072 104007
055074 013737 001172 001126
055102 013737 001226 001240
055110 042737 100100 001126
055116 023737 001124 001126
055124 001401
055126 104007
055130 000240

```

```

CMP $GDDAT,$BDDAT ;AL BITS OK ?
BEQ 86$ ;BR IF OK FROM PORT A.
EMT 7
86$: MOV $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
MOV PORTB,PTNBR ;CHANGE PORT NUMBER
BIC #ATA!VV,$BDDAT ;DON'T CHECK ATTN BIT OR VV BIT
CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
BEQ 87$ ;BR IF JK
EMT 7
87$: NOP

```

055132 000004

50\$: SCOPE

1314
1328
1329

```

*****
*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 RMD5.
*
* 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.
*****

```

```

055134
055134 005737 001300
055140 001406
055142 100002
055144 000137 003074
055150 012737 177777 001300
055156 012737 055172 001106
055164 012737 055172 001110
055172
055172 112737 000042 001102
055200 012706 001100
055204 012737 000002 001176

```

```

TST42:
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 #TEST42,$LPADR ;SETUP SCOPE LOOP ADDRESS
MOV #TEST42,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST42:
MOVB #42,$STNM ;MOVE #42 TO TEST NUMBER
MOV #STACK,SP ;LOAD THE STACK POINTER
MOV #2,$TIMES ;DO 2. ITERATIONS

```

1330
1381

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

055212 113760 001224 000010
055220 005060 000012
055224 012760 000011 000000
055232 012760 000013 000000
055240 113760 001226 000010
055246 005060 000012
055252 012760 000011 000000
055260 012760 000013 000000

```

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
CLR RMD5(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 RMD5(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

```

055266 113760 001224 000010
055274 013737 001224 001242

```

```

MOVB PORTA,RMCS2(R0) ;SELECT PORT A
MOV PORTA,SEIZPT ;STORE SEIZING PORT'S ADDRESS

```

```

055302 005060 000012          CLR    RMDS(RO)      ;WRITE RMDS
055306 013737 001226 001244  MOV    PORTB,OPPRT  ;'OPPOSITE' PORT ADDRESS

;FORCE AN ATTENTION BY SETTING ERRORS.

055314 012760 177777 000014          MOV    #-1,RMER1(RO) ;SET ERROR BITS

;START THE TIMER

055322 005037 001256          CLR    TIME         ;CLEAR THE ELAPSED TIME COUNTER
055326 012737 003720 001260  MOV    #2000.,WATCH ;SET WATCH TO 2000. MS
055334 113760 001226 000010  MOV    PORTB,RMCS2(RO) ;SELECT PORT B
055342 013737 001226 001240  MOV    PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;WAIT FOR DRIVE TO TIMEOUT

055350 005760 000012          1$:   TST    RMDS(RO)  ;WAIT FOR THE DRIVE TO BE RELEASED
055354 001004                      BNE    2$           ;BR IF DRIVE RELEASED
055356 005737 001260          TST    WATCH        ;WATCH AT ZERO ?
055362 001372                      BNE    1$           ;BR IF NOT
055364 104036                      EMT    36

055366 113760 001224 000010  2$:   MOV    PORTA,RMCS2(RO) ;SELECT PORT A
055374 013737 001224 001240  MOV    PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

055402 005037 001250          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
055406 016037 000012 001126  MOV    RMDS(RO),SBDDAT ;GET CONTENTS OF RMDS
055414 012737 000012 001122  MOV    #RMDS,SBADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055422 060037 001122          ADD    RO,SBADR     ;ADD RH/RM BASE ADDRESS
055426 012737 040000 001124  MOV    #ERR,$GDDAT  ;WHAT REGISTER SHOULD BE
055434 013737 001126 001164  MOV    SBDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
055442 042737 137777 001164  BIC    #^CERR,$TMP0 ;SAVE SPECIFIED BITS
055450 023737 001124 001164  CMP    $GDDAT,$TMP0 ;COMPARE THE BITS
055456 001414                      BEQ    66$          ;BR IF OK
055460 013737 001126 001174  MOV    SBDDAT,$TMP4 ;COPY 'BAD DATA'
055466 042737 040000 001174  BIC    #ERR,$TMP4   ;CLEAR THE MASKED BITS
055474 053737 001174 001124  BIS    $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
055502 104023                      EMT    23
055504 005137 001250          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
055510 000240          66$:  NOP

;THE ERROR REGISTER SHOULD CONTAIN 1'S

055512 005037 001250          CLR    CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
055516 016037 000014 001126  MOV    RMER1(RO),SBDDAT ;GET CONTENTS OF RMER1
055524 012737 000014 001122  MOV    #RMER1,SBADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055532 060037 001122          ADD    RO,SBADR     ;ADD RH/RM BASE ADDRESS
055536 012737 177777 001124  MOV    #177777,$GDDAT ;WHAT REGISTER SHOULD BE
055544 023737 001124 001126  CMP    $GDDAT,SBDDAT ;IS THE REGISTER OK ?
055552 001403                      BEQ    68$          ;BR IF OK
055554 104010                      EMT    10
055556 005137 001250          COM    CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
055562 000240          68$:  NOP

;THE ATTENTION BIT FOR PORT A SHOULD STILL BE SET

```

```

055564 005037 001250          CLR      CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
055570 016037 000012 001126  MOV      RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
055576 012737 000012 001122  MOV      #RMDS, $BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
055604 060037 001122          ADD      R0, $BDADR     ;ADD RH/RM BASE ADDRESS
055610 012737 100000 001124  MOV      #ATA, $GDDAT   ;WHAT REGISTER SHOULD BE
055616 013737 001126 001164  MOV      $BDDAT, $TMP0  ;MOVE REGISTER CONTENTS TO '$TMP0'
055624 042737 077777 001164  BIC      #^CATA, $TMP0  ;SAVE SPECIFIED BITS
055632 023737 001124 001164  CMP      $GDDAT, $TMP0  ;COMPARE THE BITS
055640 001414          BEQ      70$           ;BR IF OK
055642 013737 001126 001174  MOV      $BDDAT, $TMP4  ;COPY 'BAD DATA'
055650 042737 100000 001174  BIC      #ATA, $TMP4    ;CLEAR THE MASKED BITS
055656 053737 001174 001124  BIS      $TMP4, $GDDAT  ;'OR' WITH GOOD DATA FOR TYPEOUT
055664 104041          EMT      41
055666 005137 001250          COM      CKERR
055672 000240          NOP
70$:

```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

055674 005037 001254          CLR      RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
055700 012737 000012 001122  MOV      #RMDS, $BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
055706 060037 001122          ADD      R0, $BDADR    ;ADD THE I/O BASE ADDRESS
055712 012737 051700 001124  MOV      #51700, $GDDAT ;COMPARSION CONSTANT
055720 113760 001224 000010  MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A.
055726 016037 000012 001170  MOV      RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
055734 042737 024001 001170  BIC      #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
055742 013737 001170 001164  MOV      $TMP2, $TMP0   ;COPY IT INTO '$TMP0'
055750 042737 100100 001164  BIC      #ATA!VV, $TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
055756 113760 001226 000010  MOVVB   PORTB, RMCS2(R0) ;SELECT PORT B.
055764 016037 000012 001172  MOV      RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
055772 042737 024001 001172  BIC      #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
056000 013737 001172 001166  MOV      $TMP3, $TMP1   ;COPY IT INTO '$TMP1'
056006 042737 100100 001166  BIC      #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
056014 023737 001164 001166  CMP      $TMP0, $TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
056022 001006          BNE      72$           ;BR IF NOT
056024 005737 001164          TST      $TMP0
056030 001045          BNE      74$           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
056032 104046          EMT      46
056034 000137 056234          JMP      76$
056040 013737 001170 001126 72$:  MOV      $TMP2, $BDDAT  ;BYPASS THE REST OF THE CHECKS
056046 013737 001226 001240  MOV      PORTB, PTNBR   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
056054 113760 001226 000010  MOVVB   PORTB, RMCS2(R0) ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056062 005737 001164          TST      $TMP0
056066 001414          BEQ      73$           ;SEE IF STATUS EQ 0 FROM PORT A.
056070 013737 001224 001240  MOV      PORTA, PTNBR   ;BR IF ZERO
056076 013737 001172 001126  MOV      $TMP3, $BDDAT  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
056104 113760 001224 000010  MOVVB   PORTA, RMCS2(R0) ;'BAD DATA' FOR ERROR TYPE OUT
056112 005737 001166          TST      $TMP1
056116 001012          BNE      74$           ;SEE IF STATUS EG ZERO FROM PORT B.
056120 012737 177777 001254 73$:  MOV      #-1, RELERR   ;BR IF NOT
056126 012760 000011 000000  MOV      #11, RMCS1(R0) ;SET 'RELEASE ERROR' INDICATOR
056134 012760 000013 000000  MOV      #13, RMCS1(R0) ;CLEAR THE DRIVE
056142 104026          EMT      26           ;RELEASE THE DRIVE
056144 013737 001170 001126 74$:  MOV      $TMP2, $BDDAT  ;LOOK FOR BIT FAILURES WHEN RMDS READ
056152 013737 001224 001240  MOV      PORTA, PTNBR   ;CHANGE PORT NUMBER
056160 042737 100000 001126  BIC      #ATA, $BDDAT  ;DON'T CHECK THE ATTN BIT
056166 023737 001124 001126  CMP      $GDDAT, $BDDAT ;ALL BITS OK ?

```

```

056174 001401 BEQ 75$ ;BR IF OK FROM PORT A.
056176 104007 EMT 7
056200 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
056206 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
056214 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
056222 023737 001124 001126 CMP $GDDA1,$BDDAT ;SEE IF READ OK FROM PORT B.
056230 001401 BEQ 76$ ;BR IF OK
056232 104007 EMT 7
056234 000240 76$: NOP

```

;THE ATTENTION BIT FOR PORT B SHOULD NOT BE SET

```

056236 113760 001226 000010 MOVB PORTB, RMCS2(R0) ;SELECT PORT B
056244 013737 001226 001240 MOV PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
056252 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
056256 016037 000012 001126 MOV RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
056264 012737 000012 001127 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056272 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
056276 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
056302 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
056310 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
056316 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
056324 001414 BEQ 77$ ;BR IF OK
056326 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
056334 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
056342 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
056350 104052 EMT 52
056352 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
056356 000240 77$: NOP

```

;CLEAR ATTENTION BIT FOR PORT A

```

056360 113760 001224 000010 MOVB PORTA, RMCS2(R0) ;SELECT PORT #A
056366 005060 000012 CLR RMDS(R0) ;SEIZE THE DRIVE
056372 012760 000011 000000 MOV #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
056400 012760 000013 000000 MOV #13, RMCS1(R0) ;RELEASE THE DRIVE
056406 000004 3$: SCOPE ;LOOP ?

```

1382
1396
1397

```

:*****
:*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.
:*
:*****

```

```

056410 TST KYBCTL ;PERFORMING ONLY SINGLE TEST :
056410 005737 001300 BEQ 2$ ;BR IF NOT
056414 001406 BPL 1$ ;BR IF JUST ENTERED TEST
056416 100002

```


1398
1399

```

056420 000137 003074          JMP      EXEC      ;RETURN & GET NEXT TEST NUMBER
056424 012737 177777 001300 1$:      MOV      #-1,KYBCTL ;SET SINGLE TEST INDICATOR
056432 012737 056446 001106 2$:      MOV      #TEST43,$LPADR ;SETUP SCOPE LOOP ADDRESS
056440 012737 056446 001110          MOV      #TEST43,$LPERR ;SETUP ERROR LOOP ADDRESS
056446          TEST43:
056446 112737 000043 001102          MOVB     #43,$TSTNM ;MOVE #43 TO TEST NUMBER
056454 012706 001100          MOV      #STACK,SP ;LOAD THE STACK POINTER
056460 012737 000002 001176          MOV      #2.,$TIMES ;DO 2. ITERATIONS

;CLEAR ATTENTION BITS FOR BOTH PORTS

056466 113760 001224 000010          MOVB     PORTA,RMCS2(R0) ;SELECT PORT #A
056474 005060 000012          CLR      RMDS(R0) ;SEIZE THE DRIVE
056500 012760 000011 000000          MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
056506 012760 000013 000000          MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
056514 113760 001226 000010          MOVB     PORTB,RMCS2(R0) ;SELECT PORT #B
056522 005060 000012          CLR      RMDS(R0) ;SEIZE THE DRIVE THROUGH PORT 'B'
056526 012760 000011 000000          MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
056534 012760 000013 000000          MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

056542 113760 001226 000010          MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
056550 013737 001226 001242          MOV      PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
056556 005060 000012          CLR      RMDS(R0) ;WRITE RMDS
056562 013737 001224 001244          MOV      PORTA,OPPRT ;'OPPOSITE' PORT ADDRESS

;FORCE AN ATTENTION BY SETTING ERRORS.

056570 012760 177777 000014          MOV      #-1,RMER1(R0) ;SET ERROR BITS

;START THE TIMER

056576 005037 001256          CLR      TIME ;CLEAR THE ELAPSED TIME COUNTER
056602 012737 003720 001260          MOV      #2000.,WATCH ;SET WATCH TO 2000. MS
056610 113760 001224 000010          MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
056616 013737 001224 001240          MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

;WAIT FOR DRIVE TO TIMEOUT

056624 005760 000012 1$:      TST      RMDS(R0) ;WAIT FOR THE DRIVE TO BE RELEASED
056630 001004          BNE     2$ ;BR IF DRIVE RELEASED
056632 005737 001260          TST      WATCH ;WATCH AT ZERO ?
056636 001372          BNE     1$ ;BR IF NOT
056640 104036          EMT     36
056642          2$:
056642 113760 001226 000010          MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
056650 013737 001226 001240          MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT

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

056656 005037 001250          CLR      CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
056662 016037 000012 001126          MOV      RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
056670 012737 000012 001122          MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
056676 060037 001122          ADD     R0,$BDADR ;ADD RH/RM BASE ADDRESS
056702 012737 040000 001124          MOV      #ERR,$GDDAT ;WHAT REGISTER SHOULD BE
  
```



```

056710 013737 001126 001164      MOV    $BDDAT,$TMP0      ;MOVE REGISTER CONTENTS TO '$TMP0'
056716 042737 137777 001164      BIC    #^CERR,$TMP0     ;SAVE SPECIFIED BITS
056724 023737 001124 001164      CMP    $GDDAT,$TMP0     ;COMPARE THE BITS
056732 001414                      BEQ    66$              ;BR IF OK
056734 013737 001126 001174      MOV    $BDDAT,$TMP4     ;COPY 'BAD DATA'
056742 042737 040000 001174      BIC    #ERR,$TMP4       ;CLEAR THE MASKED BITS
056750 053737 001174 001124      BIS    $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
056756 104023                      EMT    23
056760 005137 001250                      COM    CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
056764 000240                      NOP
  
```

66\$:

;THE ERROR REGISTER SHOULD CONTAIN 1'S

```

056766 005037 001250                      CLR    CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
056772 016037 000014 001126      MOV    RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
057000 012737 000014 001122      MOV    #RMER1,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057006 060037 001122                      ADD    RO,$BDADR       ;ADD RH/RM BASE ADDRESS
057012 012737 177777 001124      MOV    #177777,$GDDAT  ;WHAT REGISTER SHOULD BE
057020 023737 001124 001126      CMP    $GDDAT,$BDDAT   ;IS THE REGISTER OK ?
057026 001403                      BEQ    68$              ;BR IF OK
057030 104010                      EMT    10
057032 005137 001250                      COM    CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
057036 000240                      NOP
  
```

68\$:

;THE ATTENTION BIT FOR PORT B SHOULD STILL BE SET

```

057040 005037 001250                      CLR    CKERR            ;CLEAR THE 'CHECK ERROR' INDICATOR
057044 016037 000012 001126      MOV    RMDS(RO),$BDDAT ;GET CONTENTS OF RMDS
057052 012737 000012 001122      MOV    #RMDS,$BDADR   ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057060 060037 001122                      ADD    RO,$BDADR       ;ADD RH/RM BASE ADDRESS
057064 012737 100000 001124      MOV    #ATA,$GDDAT     ;WHAT REGISTER SHOULD BE
057072 013737 001126 001164      MOV    $BDDAT,$TMP0     ;MOVE REGISTER CONTENTS TO '$TMP0'
057100 042737 077777 001164      BIC    #^CATA,$TMP0     ;SAVE SPECIFIED BITS
057106 023737 001124 001164      CMP    $GDDAT,$TMP0     ;COMPARE THE BITS
057114 001414                      BEQ    70$              ;BR IF OK
057116 013737 001126 001174      MOV    $BDDAT,$TMP4     ;COPY 'BAD DATA'
057124 042737 100000 001174      BIC    #ATA,$TMP4       ;CLEAR THE MASKED BITS
057132 053737 001174 001124      BIS    $TMP4,$GDDAT     ;'OR' WITH GOOD DATA FOR TYPEOUT
057140 104041                      EMT    41
057142 005137 001250                      COM    CKERR            ;SET THE REGISTER COMPARE ERROR INDICATOR
057146 000240                      NOP
  
```

70\$:

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

057150 005037 001254                      CLR    RELERR           ;CLEAR THE 'RELEASE ERROR ' INDICATOR
057154 012737 000012 001122      MOV    #RMDS,$BDADR   ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
057162 060037 001122                      ADD    RO,$BDADR       ;ADD THE I/O BASE ADDRESS
057166 012737 051700 001124      MOV    #51700,$GDDAT   ;COMPARISON CONSTANT
057174 113760 001224 000010      MOVB   PORTA, RMCS2(RO) ;SELECT PORT A.
057202 016037 000012 001170      MOV    RMDS(RO),$TMP2  ;GET THE DRIVE STATUS REGISTER FROM PORT A.
057210 042737 024001 001170      BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
057216 013737 001170 001164      MOV    $TMP2,$TMP0     ;COPY IT INTO '$TMP0'
057224 042737 100100 001164      BIC    #ATA!VV,$TMP0   ;CLEAR PORT DEPENDENT BITS FROM THE COPY
057232 113760 001226 000010      MOVB   PORTB, RMCS2(RO) ;SELECT PORT B.
057240 016037 000012 001172      MOV    RMDS(RO),$TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
057246 042737 024001 001172      BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
057254 013737 001172 001166      MOV    $TMP3,$TMP1     ;COPY IT INTO '$TMP1'
  
```

```

057262 042737 100100 001166 BIC #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
057270 023737 001164 001166 CMP $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
057276 001006 BNE 72$ ;BR IF NOT
057300 005737 001164 TST $TMP0 ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
057304 001045 BNE 74$ ;BR IF NOT
057306 104046 EMT 46
057310 000137 057510 JMP 76$ ;BYPASS THE REST OF THE CHECKS
057314 013737 001170 001126 72$: MOV $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
057322 013737 001226 001240 MOV PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057330 113760 001226 000010 MOVVB PORTB,RMCS2(R0) ;SELECT PORT B.
057336 005737 001164 TST $TMP0 ;SEE IF STATUS EQ 0 FROM PORT A.
057342 001414 BEQ 73$ ;BR IF ZERO
057344 013737 001224 001240 MOV PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
057352 013737 001172 001126 MOV $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
057360 113760 001224 000010 MOVVB PORTA,RMCS2(R0) ;SELECT PORT A.
057366 005737 001166 TST $TMP1 ;SEE IF STATUS EQ ZERO FROM PORT B.
057372 001012 BNE 74$ ;BR IF NOT
057374 012737 177777 001254 73$: MOV #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
057402 012760 000011 000000 MOV #11,RMCS1(R0) ;CLEAR THE DRIVE
057410 012760 000013 000000 MOV #13,RMCS1(R0) ;RELEASE THE DRIVE
057416 104026 EMT 26
057420 013737 001170 001126 74$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
057426 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
057434 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
057442 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
057450 001401 BEQ 75$ ;BR IF OK FROM PORT A.
057452 104007 EMT 7
057454 013737 001172 001126 75$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
057462 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
057470 042737 100000 001126 BIC #ATA,$BDDAT ;DON'T CHECK THE ATTN BIT
057476 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
057504 001401 BEQ 76$ ;BR IF OK
057506 104007 EMT 7
057510 000240 76$: NOP

```

;THE ATTENTION BIT FOR PORT A SHOULD NOT BE SET

```

057512 113760 001224 000010 MOVVB PORTA,RMCS2(R0) ;SELECT PORT A
057520 013737 001224 001240 MOV PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
057526 005037 001250 CLR CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
057532 016037 000012 001126 MOV RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
057540 012737 000012 001122 MOV #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
057546 060037 001122 ADD R0,$BDADR ;ADD RH/RM BASE ADDRESS
057552 005037 001124 CLR $GDDAT ;WHAT REGISTER SHOULD BE
057556 013737 001126 001164 MOV $BDDAT,$TMP0 ;MOVE REGISTER CONTENTS TO '$TMP0'
057564 042737 077777 001164 BIC #^CATA,$TMP0 ;SAVE SPECIFIED BITS
057572 023737 001124 001164 CMP $GDDAT,$TMP0 ;COMPARE THE BITS
057600 001414 BEQ 77$ ;BR IF OK
057602 013737 001126 001174 MOV $BDDAT,$TMP4 ;COPY 'BAD DATA'
057610 042737 100000 001174 BIC #ATA,$TMP4 ;CLEAR THE MASKED BITS
057616 053737 001174 001124 BIS $TMP4,$GDDAT ;'OR' WITH GOOD DATA FOR TYPEOUT
057624 104052 EMT 52
057626 005137 001250 COM CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
057632 000240 77$: NOP

```

;CLEAR ATTENTION BIT FOR PORT B

```

057634 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
057642 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE
057646 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
057654 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
057662 000004              3$:   SCOPE           ;LOOP ?
  
```

1400
1416
1417

```

:*****
:*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.
:*
:*****
  
```

```

057664 005737 001300      TST44:  TST   KYBCTL      ;PERFORMING ONLY SINGLE TEST ?
057664 001406              BEQ   2$             ;BR IF NOT
057672 100002              BPL   1$             ;BR IF JUST ENTERED TEST
057674 000137 003074      JMP   EXEC           ;RETURN & GET NEXT TEST NUMBER
057700 012737 177777 001300 1$:   MOV   #-1, KYBCTL    ;SET SINGLE TEST INDICATOR
057706 012737 057722 001106 2$:   MOV   #TEST44, $LPADR ;SETUP SCOPE LOOP ADDRESS
057714 012737 057722 001110      MOV   #TEST44, $LPERR ;SETUP ERROR LOOP ADDRESS
057722              TEST44:
057722 112737 000044 001102      MOVB  #44, $TSTNM    ;MOVE #44 TO TEST NUMBER
057730 012706 001100      MOV   #STACK, SP    ;LOAD THE STACK POINTER
057734 012737 000002 001176      MOV   #2., $TIMES   ;DO 2. ITERATIONS
  
```

1418
1453

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

057742 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT #A
057750 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE
057754 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
057762 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
057770 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
057776 005060 000012              CLR   RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
060002 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
060010 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
  
```

;SEIZE THE DRIVE THROUGH PORT A

```

060016 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
060024 013737 001224 001242      MOV   PORTA, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
060032 005060 000012              CLR   RMDS(R0)        ;WRITE RMDS
060036 013737 001226 001244      MOV   PORTB, OPPRT    ;'OPPOSITE' PORT ADDRESS
  
```

;START THE TIMER

```

060044 005037 001256          CLR      TIME      ;CLEAR THE ELAPSED TIME COUNTER
060050 012737 000764 001260  MOV      #500.,WATCH ;SET WATCH TO TIM. MS
060056 005737 001260          TST      WATCH     ;WATCH EQUAL TO ZERO
060062 001375          BNE      1$        ;BR IF NOT

;START THE TIMER

060064 005037 001256          CLR      TIME      ;CLEAR THE ELAPSED TIME COUNTER
060070 012737 003720 001260  MOV      #2000.,WATCH ;SET WATCH TO 2000. MS

;RETRIGGER THE TIMEOUT ONE-SHOT

060076 005760 000012          TST      RMDS(R0)   ;RETRIGGER THE ONE-SHOT
060102 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B
060110 013737 001226 001240  MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060116 005760 000012          TST      RMDS(R0)   ;WAIT FOR TIMEOUT
060122 001004          BNE      3$        ;BR IF TIMEOUT OCCURRED
060124 005737 001260          TST      WATCH     ;WATCH EQUAL TO ZERO ?
060130 001372          BNE      2$        ;BR IF NOT
060132 104036          EMT      36
060134 013737 001256 001276 3$:  MOV      TIME, TIMES ;SAVE THE ELAPSED TIME VALUE

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060142 005037 001254          CLR      RELERR     ;CLEAR THE 'RELEASE ERROR ' INDICATOR
060146 012737 000012 001122  MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
060154 060037 001122          ADD      R0,$BDADR  ;ADD THE I/O BASE ADDRESS
060160 012737 011700 001124  MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
060166 113760 001224 000010  MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
060174 016037 000012 001170  MOV      RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
060202 042737 024001 001170  BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
060210 013737 001170 001164  MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
060216 042737 100100 001164  BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060224 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B.
060232 016037 000012 001172  MOV      RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
060240 042737 024001 001172  BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
060246 013737 001172 001166  MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
060254 042737 100100 001166  BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
060262 023737 001164 001166  CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
060270 001006          BNE      66$       ;BR IF NOT
060272 005737 001164          TST      $TMP0     ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
060276 001045          BNE      68$       ;BR IF NOT
060300 104046          EMT      46
060302 000137 060466          JMP      70$       ;BYPASS THE REST OF THE CHECKS
060306 013737 001170 001126 66$:  MOV      $TMP2,$BDADR ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
060314 013737 001226 001240  MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060322 113760 001226 000010  MOVB     PORTB, RMCS2(R0) ;SELECT PORT B.
060330 005737 001164          TST      $TMP0     ;SEE IF STATUS EQ 0 FROM PORT A.
060334 001414          BEQ      67$       ;BR IF ZERO
060336 013737 001224 001240  MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
060344 013737 001172 001126  MOV      $TMP3,$BDADR ;'BAD DATA' FOR ERROR TYPE OUT
060352 113760 001224 000010  MOVB     PORTA, RMCS2(R0) ;SELECT PORT A.
060360 005737 001166          TST      $TMP1     ;SEE IF STATUS EQ ZERO FROM PORT B.
060364 001012          BNE      68$       ;BR IF NOT
060366 012737 177777 001254 67$:  MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
060374 012760 000011 000000  MOV      #11,RMCS1(R0) ;CLEAR THE DRIVE
060402 012760 000013 000000  MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
  
```

```

060410 104022 EMT 22
060412 013737 001170 001126 68$: MOV $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
060420 013737 001224 001240 MOV PORTA,PTNBR ;CHANGE PORT NUMBER
060426 023737 001124 001126 CMP $GDDAT,$BDDAT ;ALL BITS OK ?
060434 001401 BEQ 69$ ;BR IF OK FROM PORT A.
060436 104007 EMT 7
060440 013737 001172 001126 69$: MOV $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
060446 013737 001226 001240 MOV PORTB,PTNBR ;CHANGE PORT NUMBER
060454 023737 001124 001126 CMP $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
060462 001401 BEQ 70$ ;BR IF OK
060464 104007 EMT 7
060466 000240 70$: NOP
  
```

;CHECK THE TIME FROM RETRIGGER TO TIMEOUT

```

060470 023737 001276 001264 CMP TIMES,TIMEAP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
060476 003004 BGT 4$ ;BR IF GREATER
060500 023737 001276 001266 CMP TIMES,TIMEAM ;MEASURED TIME LESS THAN -25% TOLERANCE
060506 002001 BGE .+4 ;BR IF NOT
060510 4$: EMT 25
060510 104025 SCOPE ;LOOP ?
060512 000004
  
```

1454
1470
1471

```

*****
*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.
*
*****
  
```

```

060514 TST45: TST KYBCTL ;PERFORMING ONLY SINGLE TEST ?
060514 005737 001300 BEQ 2$ ;BR IF NOT
060520 001406 BPL 1$ ;BR IF JUST ENTERED TEST
060522 100002 JMP EXEC ;RETURN & GET NEXT TEST NUMBER
060524 000137 003074 MOV #-1,KYBCTL ;SET SINGLE TEST INDICATOR
060530 012737 177777 001300 1$: MOV #TEST45,$LPADR ;SETUP SCOPE LOOP ADDRESS
060536 012737 060552 001106 2$: MOV #TEST45,$LPERR ;SETUP ERROR LOOP ADDRESS
060544 012737 060552 001110
060552 TEST45: MOVB #45,$STNM ;MOVE #45 TO TEST NUMBER
060552 112737 000045 001102 MOV #STACK,SP ;LOAD THE STACK POINTER
060560 012706 001100 MOV #2,$TIMES ;DO 2. ITERATIONS
060564 012737 000002 001176
  
```

1472
1473

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

060572 113760 001224 000010 MOVB PORTA,RMCS2(R0) ;SELECT PORT #A
  
```

```

060600 005060 000012          CLR    RMDS(R0)          ;SEIZE THE DRIVE
060604 012760 000011 000000  MOV    #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
060612 012760 000013 000000  MOV    #13,RMCS1(R0)    ;RELEASE THE DRIVE
060620 113760 001226 000010  MOVB   PORTB,RMCS2(R0) ;SELECT PORT #B
060626 005060 000012          CLR    RMDS(R0)          ;SEIZE THE DRIVE THROUGH PORT 'B'
060632 012760 000011 000000  MOV    #11,RMCS1(R0)    ;ISSUE DRIVE CLEAR
060640 012760 000013 000000  MOV    #13,RMCS1(R0)    ;RELEASE THE DRIVE

                                ;SEIZE THE DRIVE THROUGH PORT B

060646 113760 001226 000010  MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
060654 013737 001226 001242  MOV    PORTB,SEIZPT    ;STORE SEIZING PORT'S ADDRESS
060662 005060 000012          CLR    RMDS(R0)          ;WRITE RMDS
060666 013737 001224 001244  MOV    PORTA,OPPRT     ;'OPPOSITE' PORT ADDRESS

                                ;START THE TIMER

060674 005037 001256          CLR    TIME              ;CLEAR THE ELAPSED TIME COUNTER
060700 012737 000764 001260  MOV    #500.,WATCH     ;SET WATCH TO TIM. MS
060706 005737 001260 1$:    TST    WATCH            ;WATCH EQUAL TO ZERO
060712 001375          BNE    1$                ;BR IF NOT

                                ;START THE TIMER

060714 005037 001256          CLR    TIME              ;CLEAR THE ELAPSED TIME COUNTER
060720 012737 003720 001260  MOV    #2000.,WATCH    ;SET WATCH TO 2000. MS

                                ;RETRIGGER THE TIMEOUT ONE-SHOT

060726 005760 000012          TST    RMDS(R0)          ;RETRIGGER THE ONE-SHOT
060732 113760 001224 000010  MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
060740 013737 001224 001240  MOV    PORTA,PTNBR     ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
060746 005760 000012 2$:    TST    RMDS(R0)          ;WAIT FOR TIMEOUT
060752 001004          BNE    3$                ;BR IF TIMEOUT OCCURRED
060754 005737 001260  TST    WATCH            ;WATCH EQUAL TO ZERO ?
060760 001372          BNE    2$                ;BR IF NOT
060762 104036          EMT    36                ;
060764 013737 001256 001276 3$:    MOV    TIME,TIMES        ;SAVE THE ELAPSED TIME VALUE

                                ;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

060772 005037 001254          CLR    RELERR            ;CLEAR THE 'RELEASE ERROR ' INDICATOR
060776 012737 000012 001122  MOV    #RMDS,$BDADR     ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
061004 060037 001122          ADD    R0,$BDADR        ;ADD THE I/O BASE ADDRESS
061010 012737 011700 001124  MOV    #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
061016 113760 001224 000010  MOVB   PORTA,RMCS2(R0) ;SELECT PORT A.
061024 016037 000012 001170  MOV    RMDS(R0),$TMP2   ;GET THE DRIVE STATUS REGISTER FROM PORT A.
061032 042737 024001 001170  BIC    #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
061040 013737 001170 001164  MOV    $TMP2,$TMP0      ;COPY IT INTO '$TMP0'
061046 042737 100100 001164  BIC    #ATA!VV,$TMP0    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061054 113760 001226 000010  MOVB   PORTB,RMCS2(R0) ;SELECT PORT B.
061062 016037 000012 001172  MOV    RMDS(R0),$TMP3   ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061070 042737 024001 001172  BIC    #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
061076 013737 001172 001166  MOV    $TMP3,$TMP1      ;COPY IT INTO '$TMP1'
061104 042737 100100 001166  BIC    #ATA!VV,$TMP1    ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061112 023737 001164 001166  CMP    $TMP0,$TMP1      ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
061120 001006          BNE    66$                ;BR IF NOT
  
```

```

061122 005737 001164      TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
061126 001045      BNE      68$      ;BR IF NOT
061130 1040 5      EMT      46
061132 000137 061316      JMP      70$      ;BYPASS THE REST OF THE CHECKS
061136 013737 001170 001126 66$:  MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
061144 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061152 113760 001226 000010      MOV      PORTB,RMCS2(RO) ;SELECT PORT B.
061160 005737 001164      TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
061164 001414      BEQ      67$      ;BR IF ZERO
061166 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
061174 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
061202 113760 001224 000010      MOV      PORTA,RMCS2(RO) ;SELECT PORT A.
061210 005737 001166      TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
061214 001012      BNE      68$      ;BR IF NOT
061216 012737 177777 001254 67$:  MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
061224 012760 000011 000000      MOV      #11,RMCS1(RO) ;CLEAR THE DRIVE
061232 012760 000013 000000      MOV      #13,RMCS1(RO) ;RELEASE THE DRIVE
061240 104022      EMT      22
061242 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
061250 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
061256 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
061264 001401      BEQ      69$      ;BR IF OK FROM PORT A.
061266 104007      EMT      7
061270 013737 001172 001126 69$:  MOV      $TMP3,$BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
061276 013737 001226 001240      MOV      PORTB,PTNBR ;CHANGE PORT NUMBER
061304 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
061312 001401      BEQ      70$      ;BR IF OK
061314 104007      EMT      7
061316 000240      NOP
  
```

;CHFK THE TIME FROM RETRIGGER TO TIMEOUT

```

061320 023737 001276 001272      CMP      TIMES,TIMEBP ;MEASURED TIME GREATER THAN +25% TOLERANCE ?
061326 003004      BGT      4$      ;BR IF GREATER
061330 023737 001276 001274      CMP      TIMES,TIMEBM ;MEASURED TIME LESS THAN -25% TOLERANCE
061336 002001      BGE      .+4      ;BR IF NOT
061340      EMT      25
061342 000004      SCOPE      ;LOOP ?
  
```

1474
 1495
 1496

```

:*****
:*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,$TSTNM ;MOVE #46 TO TEST NUMBER
    MOV    #STACK,SP ;LOAD THE STACK POINTER
    MOV    #2,$TIMES ;DO 2. ITERATIONS
    
```

```

061344
061344 005737 001300
061350 001406
061352 100002
061354 000137 003074
061360 012737 177777 001300
061366 012737 061402 001106
061374 012737 061402 001110
061402
061402 112737 000046 001102
061410 012706 001100
061414 012737 000002 001176
    
```

1497
1541

```

;CLEAR ATTENTION BITS FOR BOTH PORTS
    
```

```

061422 113760 001224 000010    MOVB   PORTA,RMCS2(R0) ;SELECT PORT #A
061430 005060 000012            CLR    RMDS(R0)       ;SEIZE THE DRIVE
061434 012760 000011 000000    MOV    #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
061442 012760 000013 000000    MOV    #13,RMCS1(R0) ;RELEASE THE DRIVE
061450 113760 001226 000010    MOVB   PORTB,RMCS2(R0) ;SELECT PORT #B
061456 005060 000012            CLR    RMDS(R0)       ;SEIZE THE DRIVE THROUGH PORT 'B'
061462 012760 000011 000000    MOV    #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
061470 012760 000013 000000    MOV    #13,RMCS1(R0) ;RELEASE THE DRIVE
    
```

```

;SEIZE THE DRIVE THROUGH PORT B
    
```

```

061476 113760 001226 000010    MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
061504 013737 001226 001242    MOV    PORTB,SEIZPT ;STORE SEIZING PORT'S ADDRESS
061512 005060 000012            CLR    RMDS(R0)       ;WRITE RMDS
061516 013737 001224 001244    MOV    PORTA,OPPRT   ;'OPPOSITE' PORT ADDRESS
061524 113760 001224 000010    MOVB   PORTA,RMCS2(R0) ;SELECT PORT A
061532 013737 001224 001240    MOV    PORTA,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

```

;SET REQUEST THROUGH PORT A
    
```

```

061540 005060 000012            CLR    RMDS(R0)       ;SET REQUEST FOR PORT A
061544 113760 001226 000010    MOVB   PORTB,RMCS2(R0) ;SELECT PORT B
061552 013737 001226 001240    MOV    PORTB,PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
    
```

```

;RELEASE THE DRIVE THROUGH PORT B
    
```

```

061560 012760 000013 000000    MOV    #13,RMCS1(R0) ;RELEASE DRIVE THROUGH PORT B
    
```

```

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

```

061566 013737 001264 001260    MOV    TIMEAP,WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
    
```

```

;VERIFY THAT THE DRIVE IS SEIZED BY PORT A
    
```

```

061574 005037 001250            CLR    CKERR          ;CLEAR THE 'CHECK ERROR' INDICATOR
061600 016037 000012 001126    MOV    RMDS(R0),$BDDAT ;GET CONTENTS OF RMDS
    
```



```

061606 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
061614 060037 001122              ADD      R0,$BDADR   ;ADD RH/RM BASE ADDRESS
061620 005037 001124              CLR      $GDDAT     ;WHAT REGISTER SHOULD BE
061624 023737 001124 001126      CMP      $GDDAT,$BDDAT ;IS THE REGISTER OK ?
061632 001403              BEQ      66$        ;BR IF OK
061634 104031              EMT      31
061636 005137 001250              COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
061642 000240              NOP
061644 005737 001250              TST      CKERR      ;REGISTER OK ?
061650 001402              BEQ      +5         ;BR IF OK
061652 000137 062226              JMP      1$         ;BYPASS REST OF TEST IF NOT
  
```

66\$:

```

061656 005737 001260      ;WAIT FOR THE TIMER TO RELEASE THE DRIVE
061662 001375              TST      WATCH     ;WATCH EQUAL ZERO ?
                          BNE      -4         ;BR IF NOT
  
```

;CONFIRM THAT THE DRIVE HAS TIMED OUT

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```

061664 005037 001254              CLR      RELERR     ;CLEAR THE 'RELEASE ERROR ' INDICATOR
061670 012737 000012 001122      MOV      #RMDS,$BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
061676 060037 001122              ADD      R0,$BDADR   ;ADD THE I/O BASE ADDRESS
061702 012737 011700 001124      MOV      #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
061710 113760 001224 000010      MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A.
061716 016037 000012 001170      MOV      RMDS(R0),$TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
061724 042737 024001 001170      BIC      #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
061732 013737 001170 001164      MOV      $TMP2,$TMP0 ;COPY IT INTO '$TMP0'
061740 042737 100100 001164      BIC      #ATA!VV,$TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
061746 113760 001226 000010      MOVVB   PORTB, RMCS2(R0) ;SELECT PORT B.
061754 016037 000012 001172      MOV      RMDS(R0),$TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
061762 042737 024001 001172      BIC      #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
061770 013737 001172 001166      MOV      $TMP3,$TMP1 ;COPY IT INTO '$TMP1'
061776 042737 100100 001166      BIC      #ATA!VV,$TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062004 023737 001164 001166      CMP      $TMP0,$TMP1 ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
062012 001006              BNE      68$        ;BR IF NOT
062014 005737 001164              TST      $TMP0      ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
062020 001045              BNE      70$        ;BR IF NOT
062022 104046              EMT      46
062024 000137 062224              JMP      72$        ;BYPASS THE REST OF THE CHECKS
062030 013737 001170 001126 68$:  MOV      $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
062036 013737 001226 001240      MOV      PORTB,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062044 113760 001226 000010      MOVVB   PORTB, RMCS2(R0) ;SELECT PORT B.
062052 005737 001164              TST      $TMP0      ;SEE IF STATUS EQ 0 FROM PORT A.
062056 001414              BEQ      69$        ;BR IF ZERO
062060 013737 001224 001240      MOV      PORTA,PTNBR ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062066 013737 001172 001126      MOV      $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
062074 113760 001224 000010      MOVVB   PORTA, RMCS2(R0) ;SELECT PORT A.
062102 005737 001166              TST      $TMP1      ;SEE IF STATUS EQ ZERO FROM PORT B.
062106 001012              BNE      70$        ;BR IF NOT
062110 012737 177777 001254 69$:  MOV      #-1,RELERR ;SET 'RELEASE ERROR' INDICATOR
062116 012760 000011 000000      MOV      #11,RMCS1(R0) ;CLEAR THE DRIVE
062124 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
062132 104035              EMT      35
062134 013737 001170 001126 70$:  MOV      $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
062142 013737 001224 001240      MOV      PORTA,PTNBR ;CHANGE PORT NUMBER
  
```

```
062150 042737 100000 001126      BIC      #ATA,$BDDAT      ;DON'T CHECK THE ATTN BIT
062156 023737 001124 001126      CMP      $GDDAT,$BDDAT ;ALL BITS OK ?
062164 001401                      BEQ      71$           ;BR IF OK FROM PORT A.
062166 104007                      EMT      7
062170 013737 001172 001126 71$: MOV      $TMP3,$BDDAT   ;CHECK RMDs FOR BIT FAILURES - FROM PORT B.
062176 013737 001226 001240      MOV      PORTB,PTNBR   ;CHANGE PORT NUMBER
062204 042737 100000 001126      BIC      #ATA,$BDDAT   ;DON'T CHECK THE ATTN BIT
062212 023737 001124 001126      CMP      $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
062220 001401                      BEQ      72$           ;BR IF OK
062222 104007                      EMT      7
062224 000240 72$: NOP
062226 000004 1$:  SCOPE                ;LOOP ?
```

1542
1560
1561

```
::*****
:*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.
:*
```

```
::*****
TST47:
062230 005737 001300      TST      KYBCTL        ;PERFORMING ONLY SINGLE TEST ?
062234 001406                      BEQ      2$           ;BR IF NOT
062236 100002                      BPL      1$           ;BR IF JUST ENTERED TEST
062240 000137 003074      JMP      EXEC          ;RETURN & GET NEXT TEST NUMBER
062244 012737 177777 001300 1$: MOV      #-1,KYBCTL   ;SET SINGLE TEST INDICATOR
062252 012737 062266 001106 2$: MOV      #TEST47,$LPADR ;SETUP SCOPE LOOP ADDRESS
062260 012737 062266 001110      MOV      #TEST47,$LPERR ;SETUP ERROR LOOP ADDRESS
062266                      TEST47:
062266 112737 000047 001102      MOV      #47,$TSTNM   ;MOVE #47 TO TEST NUMBER
062274 012706 001100      MOV      #STACK,SP    ;LOAD THE STACK POINTER
062300 012737 000002 001176      MOV      #2,$TIMES    ;DO 2. ITERATIONS
```

1562
1563

;CLEAR ATTENTION BITS FOR BOTH PORTS

```
062306 113760 001224 000010      MOV      PORTA,RMCS2(R0) ;SELECT PORT #A
062314 005060 000012      CLR      RMDs(R0)      ;SEIZE THE DRIVE
062320 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
062326 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
062334 113760 001226 000010      MOV      PORTB,RMCS2(R0) ;SELECT PORT #B
062342 005060 000012      CLR      RMDs(R0)      ;SEIZE THE DRIVE THROUGH PORT 'B'
062346 012760 000011 000000      MOV      #11,RMCS1(R0) ;ISSUE DRIVE CLEAR
062354 012760 000013 000000      MOV      #13,RMCS1(R0) ;RELEASE THE DRIVE
```

;SEIZE THE DRIVE THROUGH PORT A

```
062362 113760 001224 000010      MOV  PORTA, RMCS2(R0) ;SELECT PORT A
062370 013737 001224 001242      MOV  PORTA, SEIZPT ;STORE SEIZING PORT'S ADDRESS
062376 005060 000012              CLR  RMDS(R0) ;WRITE RMDS
062402 013737 001226 001244      MOV  PORTB, OFPRT ;'OPPOSITE' PORT ADDRESS
062410 113760 001226 000010      MOV  PORTB, RMCS2(R0) ;SELECT PORT B
062416 013737 001226 001240      MOV  PORTB, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

;SET REQUEST THROUGH PORT B

```
062424 005060 000012              CLR  RMDS(R0) ;SET REQUEST FOR PORT B
062430 113760 001224 000010      MOV  PORTA, RMCS2(R0) ;SELECT PORT A
062436 013737 001224 001240      MOV  PORTA, PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
```

;RELEASE THE DRIVE THROUGH PORT A

```
062444 012760 000013 000000      MOV  #13, RMCS1(R0) ;RELEASE DRIVE THROUGH PORT A
```

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

```
062452 013737 001272 001260      MOV  TIMEBP, WATCH ;SET WATCH TO MEASURED TIMEOUT VALUE + 25%
```

;VERIFY THAT THE DRIVE IS SEIZED BY PORT B

```
062460 005037 001250              CLR  CKERR ;CLEAR THE 'CHECK ERROR' INDICATOR
062464 016037 000012 001126      MOV  RMDS(R0), $BDDAT ;GET CONTENTS OF RMDS
062472 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
062500 060037 001122              ADD  R0, $BDADR ;ADD RH/RM BASE ADDRESS
062504 005037 001124              CLR  $GDDAT ;WHAT REGISTER SHOULD BE
062510 023737 001124 001126      CMP  $GDDAT, $BDDAT ;IS THE REGISTER OK ?
062516 001403              BEQ  66$ ;BR IF OK
062520 104031              EMT  31
062522 005137 001250              COM  CKERR ;SET THE REGISTER COMPARE ERROR INDICATOR
062526 000240 66$:              NOP
062530 005737 001250              TST  CKERR ;REGISTER OK ?
062534 001402              BEQ  +6 ;BR IF OK
062536 000137 063112              JMP  1$ ;BYPASS REST OF TEST IF NOT
```

;WAIT FOR THE TIMER TO RELEASE THE DRIVE

```
062542 005737 001260              TST  WATCH ;WATCH EQUAL ZERO ?
062546 001375              BNE  -4 ;BR IF NOT
```

;CONFIRM THAT THE DRIVE HAS TIMED OUT

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
062550 005037 001254              CLR  RELERR ;CLEAR THE 'RELEASE ERROR' INDICATOR
062554 012737 000012 001122      MOV  #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
062562 060037 001122              ADD  R0, $BDADR ;ADD THE I/O BASE ADDRESS
062566 012737 011700 001124      MOV  #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
062574 113760 001224 000010      MOV  PORTA, RMCS2(R0) ;SELECT PORT A.
062602 016037 000012 001170      MOV  RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
062610 042737 024001 001170      BIC  #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
062616 013737 001170 001164      MOV  $TMP2, $TMP0 ;COPY IT INTO '$TMP0'
062624 042737 100100 001164      BIC  #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
```

```

062632 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
062640 016037 000012 001172      MOV   RMDS(R0), $TMP3  ;GET THE DRIVE STATUS REGISTER FROM PORT B.
062646 042737 024001 001172      BIC   #PIP!WRL.OM, $TMP3 ;CLEAR DONT CARES
062654 013737 001172 001166      MOV   $TMP3, $TMP1    ;COPY IT INTO '$TMP1'
062662 042737 100100 001166      BIC   #ATA!VV, $TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
062670 023737 001164 001166      CMP   $TMP0, $TMP1    ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
062676 001006                BNE   68$             ;BR IF NOT
062700 005737 001164                TST   $TMP0           ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
062704 001045                BNE   70$             ;BR IF NOT
062706 104046                EMT   46
062710 000137 063110                JMP   72$             ;BYPASS THE REST OF THE CHECKS
062714 013737 001170 001126 68$:  MOV   $TMP2, $BDDAT   ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
062722 013737 001226 001240      MOV   PORTB, PTNBR    ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
062730 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
062736 005737 001164                TST   $TMP0           ;SEE IF STATUS EQ 0 FROM PORT A.
062742 001414                BEQ   69$             ;BR IF ZERO
062744 013737 001224 001240      MOV   PORTA, PTNBR    ;SEIZING PORT , TEST SHOWS DRIVE NOT IN NEUTRAL
062752 013737 001172 001126      MOV   $TMP3, $BDDAT   ;'BAD DATA' FOR ERROR TYPE OUT
062760 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
062766 005737 001166                TST   $TMP1           ;SEE IF STATUS EQ ZERO FROM PORT B.
062772 001012                BNE   70$             ;BR IF NOT
062774 012737 177777 001254 69$:  MOV   #-1, RELERR     ;SET 'RELEASE ERROR' INDICATOR
063002 012760 000011 000000      MOV   #11, RMCS1(R0)  ;CLEAR THE DRIVE
063010 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE
063016 104035                EMT   35
063020 013737 001170 001126 70$:  MOV   $TMP2, $BDDAT   ;LOOK FOR BIT FAILURES WHEN RMDS READ
063026 013737 001224 001240      MOV   PORTA, PTNBR    ;CHANGE PORT NUMBER
063034 042737 100000 001126      BIC   #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
063042 023737 001124 001126      CMP   $GDDAT, $BDDAT ;ALL BITS OK ?
063050 001401                BEQ   71$             ;BR IF OK FROM PORT A.
063052 104007                EMT   7
063054 013737 001172 001126 71$:  MOV   $TMP3, $BDDAT   ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
063062 013737 001226 001240      MOV   PORTB, PTNBR    ;CHANGE PORT NUMBER
063070 042737 100000 001126      BIC   #ATA, $BDDAT    ;DON'T CHECK THE ATTN BIT
063076 023737 001124 001126      CMP   $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
063104 001401                BEQ   72$             ;BR IF OK
063106 104007                EMT   7
063110 000240                NOP
063112 000004                1$:  SCOPE                ;LOOP ?
  
```

1564
1589
1590

```

:*****
:*TEST 50      PORT 'A' SEIZE ACCESS TEST
:*
:*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.

063114
 063114 005737 001300
 063120 001406
 063122 100002
 063124 000137 003074
 063130 012737 177777 001300
 063136 012737 063152 001106
 063144 012737 063152 001110
 063152
 063152 112737 000050 001102
 063160 012706 001100
 063164 012737 000012 001176

1591
 1630

```
TST50:
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      #TEST50,$LPADR ;SETUP SCOPE LOOP ADDRESS
          MOV      #TEST50,$LPERR ;SETUP ERROR LOOP ADDRESS
TEST50:
          MOVB     #50,$STSTNM ;MOVE #50 TO TEST NUMBER
          MOV      #STACK,SP ;LOAD THE STACK POINTER
          MOV      #10,$TIMES ;DO 10. ITERATIONS
```

;CLEAR ATTENTION BITS FOR BOTH PORTS

063172 113760 001224 000010
 063200 005060 000012
 063204 012760 000011 000000
 063212 012760 000013 000000
 063220 113760 001226 000010
 063226 005060 000012
 063232 012760 000011 000000
 063240 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

063246 113760 001224 000010
 063254 013737 001224 001242
 063262 005060 000012
 063266 013737 001226 001244
 063274 012760 177777 000014
 063302 012760 177777 000042
 063310 113760 001226 000010
 063316 013737 001226 001240
 063324 004737 064164
 063330 113760 001224 000010
 063336 013737 001224 001240
 063344 005060 000042
 063350 005060 000014
 063354 013760 001236 000016
 063362 113760 001226 000010
 063370 013737 001226 001240
 063376 012760 177777 000014
 063404 012760 177777 000042
 063412 113760 001224 000010
 063420 013737 001224 001240
 063426 004737 064164

```
          MOVB     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
          MOV      #-1,RMER1(R0) ;LOAD 1'S INTO RMER1 THROUGH PORT A
          MOV      #-1,RMER2(R0) ;LOAD 1'S INTO RMER2 THROUGH PORT A
          MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
          MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
          JSR      PC,TST50B ;CHECK THE REGISTERS THROUGH PORT B
          MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
          MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
          CLR      RMER2(R0) ;CLEAR RMER2 ON PORT A
          CLR      RMER1(R0) ;CLEAR RMER1 ON PORT A
          MOV      ASR1,RMAS(R0) ;CLEAR THE ATTENTION BIT FOR PORT A
          MOVB     PORTB,RMCS2(R0) ;SELECT PORT B
          MOV      PORTB,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
          MOV      #-1,RMER1(R0) ;LOAD 1'S INTO RMER1 THROUGH PORT B
          MOV      #-1,RMER2(R0) ;LOAD 1'S INTO RMER2 THROUGH PORT B
          MOVB     PORTA,RMCS2(R0) ;SELECT PORT A
          MOV      PORTA,PTNBR ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
          JSR      PC,TST50B ;CHECK THE REGISTERS THROUGH PORT A
```

;RELEASE THE DRIVE FROM PORT A

```
063432 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
063440 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063446 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

```
063454 005037 001254              CLR   RELERR        ;CLEAR 'RELEASE ERROR' INDICATOR
063460 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
063466 012737 000012 001122      MOV   #RMDS,$BDADR  ;REGISTER ADDRESS INCREMENT
063474 060037 001122              ADD   R0,$BDADR    ;REGISTER BASE ADDRESS FOR TYPEOUT
063500 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
063506 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063514 016037 000012 001164      MOV   RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT B
063522 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
063530 013737 001224 001240      MOV   PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063536 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT A
063544 001404                      BEQ   66$           ;BR IF STATUS FROM PORT A ZERO
063546 005737 001164              TST   $TMP0        ;IS STATUS FROM PORT B ZERO ?
063552 001401                      BEQ   66$           ;BR IF ZERO
063554 104031                      EMT   31
063556 013737 001164 001126 66$: MOV   $TMP0,$BDDAT  ;CHECK STATUS FROM PORT B
063564 013737 001226 001240      MOV   PORTB, PTNBR  ;CHANGE PORT ADDRESS FOR TYPEOUT
063572 023737 001124 001126      CMP   $GDDAT,$BDDAT ;COMPARE WITH CONSTANT
063600 001401                      BEQ   67$           ;BR IF OK
063602 104027                      EMT   27
063604 000240 67$: NOP
```

;RELEASE THE DRIVE FROM PORT B

```
063606 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
063614 013737 001226 001240      MOV   PORTB, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
063622 012760 000013 000000      MOV   #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT B
```

;VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
063630 005037 001254              CLR   RELERR        ;CLEAR THE 'RELEASE ERROR' INDICATOR
063634 012737 000012 001122      MOV   #RMDS,$BDADR  ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
063642 060037 001122              ADD   R0,$BDADR    ;ADD THE I/O BASE ADDRESS
063646 012737 011700 001124      MOV   #MOL!PGM!DPR!DRY!VV,$GDDAT ;COMPARISON CONSTANT
063654 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A.
063662 016037 000012 001170      MOV   RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
063670 042737 024001 001170      BIC   #PIP!WRL!OM,$TMP2 ;CLEAR DONT CARES
063676 013737 001170 001164      MOV   $TMP2,$TMP0   ;COPY IT INTO '$TMP0'
063704 042737 100100 001164      BIC   #ATA!VV,$TMP0  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063712 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B.
063720 016037 000012 001172      MOV   RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
063726 042737 024001 001172      BIC   #PIP!WRL!OM,$TMP3 ;CLEAR DONT CARES
063734 013737 001172 001166      MOV   $TMP3,$TMP1   ;COPY IT INTO '$TMP1'
063742 042737 100100 001166      BIC   #ATA!VV,$TMP1  ;CLEAR PORT DEPENDENT BITS FROM THE COPY
063750 023737 001164 001166      CMP   $TMP0,$TMP1   ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
063756 001006                      BNE   68$           ;BR IF NOT
063760 005737 001164              TST   $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
063764 001045                      BNE   70$           ;BR IF NOT
063766 104046                      EMT   46
```

```

063770 000137 064154          JMP      72$          ;BYPASS THE REST OF THE CHECKS
063774 013737 001170 001126 68$:  MOV     $TMP2,$BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
064002 013737 001226 001240      MOV     PORTB,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
064010 113760 001226 000010      MOVVB  PORTB,RMCS2(RO) ;SELECT PORT B.
064016 005737 001164          TST     $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
064022 001414          BEQ     69$          ;BR IF ZERO
064024 013737 001224 001240      MOV     PORTA,PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
064032 013737 001172 00 126      MOV     $TMP3,$BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
064040 113760 001224 000010      MOVVB  PORTA,RMCS2(RO) ;SELECT PORT A.
064046 005737 001166          TST     $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
064052 001012          BNE     70$          ;BR IF NOT
064054 012737 177777 001254 69$:  MOV     #-1,RELERR   ;SET 'RELEASE ERROR' INDICATOR
064062 012760 000011 000000      MOV     #11,RMCS1(RO) ;CLEAR THE DRIVE
064070 012760 000013 000000      MOV     #13,RMCS1(RO) ;RELEASE THE DRIVE
064076 104026          EMT     26
064100 013737 001170 001126 70$:  MOV     $TMP2,$BDDAT ;LOOK FOR BIT FAILURES WHEN RMD5 READ
064106 013737 001224 001240      MOV     PORTA,PTNBR  ;CHANGE PORT NUMBER
064114 023737 001124 001126      CMP     $GDDAT,$BDDAT ;ALL BITS OK ?
064122 001401          BEQ     71$          ;BR IF OK FROM PORT A.
064124 104007          EMT     7
064126 013737 001172 001126 71$:  MOV     $TMP3,$BDDAT ;CHECK RMD5 FOR BIT FAILURES - FROM PORT B.
064134 013737 001226 001240      MOV     PORTB,PTNBR  ;CHANGE PORT NUMBER
064142 023737 001124 001126      CMP     $GDDAT,$BDDAT ;SEE IF READ OK FROM PORT B.
064150 001401          BEQ     72$          ;BR IF OK
064152 104007          EMT     7
064154 000240          NOP
064156 000004          SCOPE
1631 064160 000137 064406      JMP     TST51        ;LOOP ?
                                ;GO TO THE NEXT TEST
    
```

;CHECK THE REGISTERS ON THE SELECTED PORT

```

064164          TST50B:
064164 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
064170 016037 000014 001126      MOV     RMER1(RO),$BDDAT ;GET CONTENTS OF RMER1
064176 012737 000014 001122      MOV     #RMER1,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
064204 060037 001122          ADD     RO,$BDADR    ;ADD RH/RM BASE ADDRESS
064210 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
064214 023737 001124 001126      CMP     $GDDAT,$BDDAT ;IS THE REGISTER OK ?
064222 001403          BEQ     64$        ;BR IF OK
064224 104006          EMT     6
064226 005137 001250          COM     CKERR        ;SET THE REGISTER COMPARE ERROR INDICATOR
064232 016037 000000 001126 64$:  MOV     RMCS1(RO),$BDDAT ;GET THE CONTENTS OF RMCS1
064240 012737 000000 001122      MOV     #RMCS1,$BDADR ;FORM ADDRESS OF REGISTER
064246 060037 001122          ADD     RO,$BDADR    ;ADDRESS BASE
064252 032737 020000 001126      BIT     #MCPE,$BDDAT ;IS 'MCPE' SET ?
064260 001404          BEQ     65$        ;BR IF NOT
064262 104011          EMT     11
064264 012760 040000 000000      MOV     #TRE, RMCS1(RO) ;CLEAR 'MCPE'
064272 000240          NOP
064274 005037 001250          CLR     CKERR        ;CLEAR THE 'CHECK ERROR' INDICATOR
064300 016037 000042 001126      MOV     RMER2(RO),$BDDAT ;GET CONTENTS OF RMER2
064306 012737 000042 001122      MOV     #RMER2,$BDADR ;FORM REGISTER ADDRESS OF ERROR MESSAGE
064314 060037 001122          ADD     RO,$BDADR    ;ADD RH/RM BASE ADDRESS
064320 005037 001124          CLR     $GDDAT      ;WHAT REGISTER SHOULD BE
064324 023737 001124 001126      CMP     $GDDAT,$BDDAT ;IS THE REGISTER OK ?
064332 001403          BEQ     66$        ;BR IF OK
064334 104006          EMT     6
    
```



```

064336 005137 001250          COM      CKERR          ;SET THE REGISTER COMPARE ERROR INDICATOR
064342 016037 000000 001126 66$:    MOV      RMCS1(R0), $BDDAT ;GET THE CONTENTS OF RMCS1
064350 012737 000000 001122          MOV      #RMCS1, $BDADR  ;FORM ADDRESS OF REGISTER
064356 060037 001122          ADD      R0, $BDADR      ;ADDRESS BASE
064362 032737 020000 001126          BIT      #MCPE, $BDDAT  ;IS 'MCPE' SET ?
064370 001404          BEQ      67$            ;BR IF NOT
064372 104011          EMT      11
064374 012760 040000 000000          MOV      #TRE, RMCS1(R0) ;CLEAR 'MCPE'
064402 000240          67$:    NOP
064404 000207          RTS      PC            ;RETURN
  
```

1632
 1657
 1658

```

:*****
: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.
:*****
  
```

```

064406          TST51:
064406 005737 001300          TST      KYBCTL          ;PERFORMING ONLY SINGLE TEST ?
064412 001406          BEQ      2$              ;BR IF NOT
064414 100002          BPL      1$              ;BR IF JUST ENTERED TEST
064416 000137 003074          JMP      EXEC            ;RETURN & GET NEXT TEST NUMBER
064422 012737 177777 001300 1$:    MOV      #-1, KYBCTL     ;SET SINGLE TEST INDICATOR
064430 012737 064444 001106 2$:    MOV      #TEST51, $LPADR ;SETUP SCOPE LOOP ADDRESS
064436 012737 064444 001110          MOV      #TEST51, $LPERR ;SETUP ERROR LOOP ADDRESS
064444          TEST51:
064444 112737 000051 001102          MOV#B   #51, $STSTM      ;MOVE #51 TO TEST NUMBER
064452 012706 001100          MOV      #STACK, SP     ;LOAD THE STACK POINTER
064456 012737 000012 001176          MOV      #10., $TIMES   ;;DO 10. ITERATIONS
  
```

1659
 1660

;CLEAR ATTENTION BITS FOR BOTH PORTS

```

064464 113760 001224 000010          MOV#B   PORTA, RMCS2(R0) ;SELECT PORT #A
064472 005060 000012          CLR      RMDS(R0)       ;SEIZE THE DRIVE
064476 012760 000011 000000          MOV      #11, RMCS1(R0) ;ISSUE DRIVE CLEAR
064504 012760 000013 000000          MOV      #13, RMCS1(R0) ;RELEASE THE DRIVE
  
```



```

064512 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT #B
064520 005060 000012 000000      CLR   RMDS(R0)        ;SEIZE THE DRIVE THROUGH PORT 'B'
064524 012760 000011 000000      MOV   #11, RMCS1(R0)  ;ISSUE DRIVE CLEAR
064532 012760 000013 000000      MOV   #13, RMCS1(R0)  ;RELEASE THE DRIVE

;SEIZE THE DRIVE THROUGH PORT B

064540 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
064546 013737 001226 001242      MOV   PORTB, SEIZPT   ;STORE SEIZING PORT'S ADDRESS
064554 005060 000012 000000      CLR   RMDS(R0)        ;WRITE RMDS
064560 013737 001224 001244      MOV   PORTA, OPPRT    ;'OPPOSITE' PORT ADDRESS
064566 012760 177777 000014      MOV   #-1, RMER1(R0)  ;LOAD 1'S INTO RMER1 THROUGH PORT B
064574 012760 177777 000042      MOV   #-1, RMER2(R0)  ;LOAD 1'S INTO RMER2 THROUGH PORT B
064602 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
064610 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064616 004737 065456 000000      JSR   PC, TST51B     ;CHECK THE REGISTERS THROUGH PORT A
064622 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
064630 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064636 005060 000042 000000      CLR   RMER2(R0)      ;CLEAR RMER2 ON PORT B
064642 005060 000014 000000      CLR   RMER1(R0)      ;CLEAR RMER1 ON PORT B
064646 013760 001236 000016      MOV   ASR1, RMAS(R0)  ;CLEAR THE ATTENTION BIT FOR PORT B
064654 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
064662 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064670 012760 177777 000014      MOV   #-1, RMER1(R0)  ;LOAD 1'S INTO RMER1 THROUGH PORT A
064676 012760 177777 000042      MOV   #-1, RMER2(R0)  ;LOAD 1'S INTO RMER2 THROUGH PORT A
064704 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
064712 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064720 004737 065456 000000      JSR   PC, TST51B     ;CHECK THE REGISTERS THROUGH PORT B

;RELEASE THE DRIVE FROM PORT B

064724 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
064732 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
064740 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

064746 005037 001254 000000      CLR   RELERR         ;CLEAR 'RELEASE ERROR' INDICATOR
064752 012737 111700 001124      MOV   #ATA!MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
064760 012737 000012 001122      MOV   #RMDS, $BDADR   ;REGISTER ADDRESS INCREMENT
064766 060037 001122 000000      ADD   R0, $BDADR      ;REGISTER BASE ADDRESS FOR TYPEOUT
064772 113760 001224 000010      MOVB  PORTA, RMCS2(R0) ;SELECT PORT A
065000 013737 001224 001240      MOV   PORTA, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065006 016037 000012 001164      MOV   RMDS(R0), $TMP0 ;READ STATUS REGISTER FROM PORT A
065014 113760 001226 000010      MOVB  PORTB, RMCS2(R0) ;SELECT PORT B
065022 013737 001226 001240      MOV   PORTB, PTNBR    ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065030 016037 000012 001126      MOV   RMDS(R0), $BDDAT ;DRIVE STATUS FROM PORT B
065036 001404 000000 000000      BEQ   66$            ;BR IF STATUS FROM PORT B ZERO
065040 005737 001164 000000      TST   $TMP0          ;IS STATUS FROM PORT A ZERO ?
065044 001401 000000 000000      BEQ   66$            ;BR IF ZERO
065046 104031 000000 000000      EMT   31
065050 013737 001164 001126 66$:  MOV   $TMP0, $BDDAT   ;CHECK STATUS FROM PORT A
065056 013737 001224 001240      MOV   PORTA, PTNBR    ;CHANGE PORT ADDRESS FOR TYPEOUT
065064 023737 001124 001126      CMP   $GDDAT, $BDDAT ;COMPARE WITH CONSTANT
065072 001401 000000 000000      BEQ   67$            ;BR IF OK
065074 104027 000000 000000      EMT   27
065076 000240 000000 000000 67$:  NOP

```

:RELEASE THE DRIVE FROM PORT A

```
065100 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A
065106 013737 001224 001240      MOV    PORTA, PTNBR  ;MOVE PORT ADDRESS TO LOCATION FOR TYPEOUT
065114 012760 000013 000000      MOV    #13, RMCS1(R0) ;ISSUE RELEASE THROUGH PORT A
```

:VERIFY THAT THE DRIVE IS STILL IN NEUTRAL

```
065122 005037 001254                CLR    RELERR        ;CLEAR THE 'RELEASE ERROR ' INDICATOR
065126 012737 000012 001122      MOV    #RMDS, $BDADR ;FORM THE ADDRESS OF RMDS FOR TYPEOUT
065134 060037 001122                ADD    R0, $BDADR   ;ADD THE I/O BASE ADDRESS
065140 012737 011700 001124      MOV    #MOL!PGM!DPR!DRY!VV, $GDDAT ;COMPARISON CONSTANT
065146 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
065154 016037 000012 001170      MOV    RMDS(R0), $TMP2 ;GET THE DRIVE STATUS REGISTER FROM PORT A.
065162 042737 024001 001170      BIC    #PIP!WRL!OM, $TMP2 ;CLEAR DONT CARES
065170 013737 001170 001164      MOV    $TMP2, $TMP0  ;COPY IT INTO '$TMP0'
065176 042737 100100 001164      BIC    #ATA!VV, $TMP0 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065204 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
065212 016037 000012 001172      MOV    RMDS(R0), $TMP3 ;GET THE DRIVE STATUS REGISTER FROM PORT B.
065220 042737 024001 001172      BIC    #PIP!WRL!OM, $TMP3 ;CLEAR DONT CARES
065226 013737 001172 001166      MOV    $TMP3, $TMP1  ;COPY IT INTO '$TMP1'
065234 042737 100100 001166      BIC    #ATA!VV, $TMP1 ;CLEAR PORT DEPENDENT BITS FROM THE COPY
065242 023737 001164 001166      CMP    $TMP0, $TMP1  ;IS THE STATUS REGISTER THE SAME FROM BOTH PORTS ?
065250 001006                        BNE    68$          ;BR IF NOT
065252 005737 001164                TST    $TMP0        ;REGISTERS ARE THE SAME: ARE THEY ZERO ?
065256 001045                        BNE    70$          ;BR IF NOT
065260 104046                        EMT    46
065262 000137 065446                JMP    72$          ;BYPASS THE REST OF THE CHECKS
065266 013737 001170 001126 68$:  MOV    $TMP2, $BDDAT ;SET UP POSSIBLE BAD DATA FOR ERROR MESSAGE
065274 013737 001226 001240      MOV    PORTB, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065302 113760 001226 000010      MOVB   PORTB, RMCS2(R0) ;SELECT PORT B.
065310 005737 001164                TST    $TMP0        ;SEE IF STATUS EQ 0 FROM PORT A.
065314 001414                        BEQ    69$          ;BR IF ZERO
065316 013737 001224 001240      MOV    PORTA, PTNBR  ;SEIZING PORT IF TEST SHOWS DRIVE NOT IN NEUTRAL
065324 013737 001172 001126      MOV    $TMP3, $BDDAT ;'BAD DATA' FOR ERROR TYPE OUT
065332 113760 001224 000010      MOVB   PORTA, RMCS2(R0) ;SELECT PORT A.
065340 005737 001166                TST    $TMP1        ;SEE IF STATUS EQ ZERO FROM PORT B.
065344 001012                        BNE    70$          ;BR IF NOT
065346 012737 177777 001254 69$:  MOV    #-1, RELERR   ;SET 'RELEASE ERROR' INDICATOR
065354 012760 000011 000000      MOV    #11, RMCS1(R0) ;CLEAR THE DRIVE
065362 012760 000013 000000      MOV    #13, RMCS1(R0) ;RELEASE THE DRIVE
065370 104026                        EMT    26
065372 013737 001170 001126 70$:  MOV    $TMP2, $BDDAT ;LOOK FOR BIT FAILURES WHEN RMDS READ
065400 013737 001224 001240      MOV    PORTA, PTNBR  ;CHANGE PORT NUMBER
065406 023737 001124 001126      CMP    $GDDAT, $BDDAT ;ALL BITS OK ?
065414 001401                        BEQ    71$          ;BR IF OK FROM PORT A.
065416 104007                        EMT    7
065420 013737 001172 001126 71$:  MOV    $TMP3, $BDDAT ;CHECK RMDS FOR BIT FAILURES - FROM PORT B.
065426 013737 001226 001240      MOV    PORTB, PTNBR  ;CHANGE PORT NUMBER
065434 023737 001124 001126      CMP    $GDDAT, $BDDAT ;SEE IF READ OK FROM PORT B.
065442 001401                        BEQ    72$          ;BR IF OK
065444 104007                        EMT    7
065446 000240                        NOP
065450 000004                        SCOPE
1661 065452 000137 065700      JMP    TST52        ;LOOP ?
                                ;GO TO THE NEXT TEST
```

;CHECK THE REGISTERS ON THE SELECTED PORT

```

065456          TST51B:
065456 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
065462 016037 000014 001126  MOV      RMER1(R0), $BDDAT ;GET CONTENTS OF RMER1
065470 012737 000014 001122  MOV      #RMER1, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
065476 060037 001122          ADD      R0, $BDADR    ;ADD RH/RM BASE ADDRESS
065502 005037 001124          CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
065506 023737 001124 001126  CMP      $GDDAT, $BDDAT ;IS THE REGISTER OK ?
065514 001403          BEQ      64$        ;BR IF OK
065516 104006          EMT      6
065520 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
065524 016037 000000 001126 64$:  MOV      RMCS1(R0), $BDDAT ;GET THE CONTENTS OF RMCS1
065532 012737 000000 001122  MOV      #RMCS1, $BDADR  ;FORM ADDRESS OF REGISTER
065540 060037 001122          ADD      R0, $BDADR    ;ADDRESS BASE
065544 032737 020000 001126  BIT      #MCPE, $BDDAT  ;IS 'MCPE' SET ?
065552 001404          BEQ      65$        ;BR IF NOT
065554 104011          EMT      11
065556 012760 040000 000000 65$:  MOV      #TRE, RMCS1(R0) ;CLEAR 'MCPE'
065564 000240          NOP
065566 005037 001250          CLR      CKERR      ;CLEAR THE 'CHECK ERROR' INDICATOR
065572 016037 000042 001126  MOV      RMER2(R0), $BDDAT ;GET CONTENTS OF RMER2
065600 012737 000042 001122  MOV      #RMER2, $BDADR  ;FORM REGISTER ADDRESS OF ERROR MESSAGE
065606 060037 001122          ADD      R0, $BDADR    ;ADD RH/RM BASE ADDRESS
065612 005037 001124          CLR      $GDDAT      ;WHAT REGISTER SHOULD BE
065616 023737 001124 001126  CMP      $GDDAT, $BDDAT ;IS THE REGISTER OK ?
065624 001403          BEQ      66$        ;BR IF OK
065626 104006          EMT      6
065630 005137 001250          COM      CKERR      ;SET THE REGISTER COMPARE ERROR INDICATOR
065634 016037 000000 001126 66$:  MOV      RMCS1(R0), $BDDAT ;GET THE CONTENTS OF RMCS1
065642 012737 000000 001122  MOV      #RMCS1, $BDADR  ;FORM ADDRESS OF REGISTER
065650 060037 001122          ADD      R0, $BDADR    ;ADDRESS BASE
065654 032737 020000 001126  BIT      #MCPE, $BDDAT  ;IS 'MCPE' SET ?
065662 001404          BEQ      67$        ;BR IF NOT
065664 104011          EMT      11
065666 012760 040000 000000 67$:  MOV      #TRE, RMCS1(R0) ;CLEAR 'MCPE'
065674 000240          NOP
065676 000207          RTS      PC      ;RETURN
    
```

1662
 1663
 1664
 1665

065700 000004

```

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

.SBTTL END OF PASS ROUTINE

```

*****
*INCREMENT THE PASS NUMBER ($PASS)
*TYPE 'END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYYY'
*WHERE XXXXX AND YYYYYY ARE DECIMAL NUMBERS
*IF THERES A MONITOR GO TO IT
*IF THERE ISN'T JUMP TO TST1AA
    
```

```

065702          $EOP:
065702 005737 001300      TST      KYBCTL      ;ENTERED TEST VIA KEYBOARD COMMAND ?
065706 001402          BEQ      .+6          ;BR IF NOT
065710 000137 003074      JMP      EXEC      ;RETURN TO KEYBOARD CONTROL
065714 005037 001102      CLR      $TSTNM     ;ZERO THE TEST NUMBER
065720 005037 001176      CLR      $TIMES     ;ZERO THE NUMBER OF ITERATIONS
065724 005237 001100      INC      $PASS      ;INCREMENT THE PASS NUMBER
065730 042737 100000 001100 BIC      #100000,$PASS ;DON'T ALLOW A NEG. NUMBER
065736 005327          DEC      (PC)+      ;LOOP?
065740 000001          $EOPCT: .WORD    1
065742 003066          BGT      $DOAGN     ;:YES
065744 012737          MOV      (PC)+,@(PC)+ ;:RESTORE COUNTER
065746 000001          $ENDCT: .WORD    1
065750 065740          $EOPCT
065752 104401 065760      TYPE     ,65$      ;:TYPE ASCIZ STRING
065756 000407          BR       64$      ;:GET OVER THE ASCIZ
;:65$: .ASCIZ <12><15>/END PASS #/
64$:
065776          MOV      $PASS,-(SP) ;:SAVE $PASS FOR TYPEOUT
065776 013746 001100      ;:TYPE PASS NUMBER
;:GO TYPE--DECIMAL ASCII WITH SIGN
066002 104405          TYPDS
066004 005737 001112      TST      $ERTTL     ;:SEE IF ANY ERRORS THIS PASS
066010 001431          BEQ      $GT42P    ;:BR IF NO ERRORS TO REPORT
066012 104401 066020      TYPE     ,67$      ;:TYPE ASCIZ STRING
066016 000421          BR       66$      ;:GET OVER THE ASCIZ
;:67$: .ASCIZ / TOTAL ERRORS SINCE LAST REPORT /
66$:
066062          MOV      $ERTTL,-(SP) ;:SAVE $ERTTL FOR TYPEOUT
066062 013746 001112      ;:TOTAL NUMBER OF ERRORS
;:GO TYPE--DECIMAL ASCII WITH SIGN
066066 104405          TYPDS
066070 005037 001112      CLR      $ERTTL     ;:CLEAR ERROR TOTAL
066074 104401 001207      $GT42P: TYPE     ,$CRLF ;:TYPE CARRIAGE RETURN, LINE FEED
066100 013700 000042      $GET42: MOV      @#42,R0 ;:GET MONITOR ADDRESS
066104 001405          BEQ      $DOAGN     ;:BRANCH IF NO MONITOR
066106 000005          RESET ;:CLEAR THE WORLD
066110 004710          $ENDAD: JSR     PC,(R0) ;:GO TO MONITOR
066112 000240          NOP ;:SAVE ROOM
066114 000240          NOP ;:FOR
066116 000240          NOP ;:ACT11
066120          $DOAGN:
066120 000137          JMP      @(PC)+      ;:RETURN
066122 003356          $RTNAD: .WORD    TST1AA
066124 377 377 000 $ENULL: .BYTE   -1,-1,0 ;:NULL CHARACTER STRING
                                .EVEN
    
```

.SBTTL CLOCK SUBROUTINES

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

;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
;IF CLOCK IS PRESENT, THE CLOCK WILL BE STARTED

```

066130 012737 066200 000004 CKCLK: MOV #CKCLK1,@#ERRVEC ;SET UP VECTOR FOR CLOCK CHECK
066136 005037 000006 CLR @#ERRVEC+2 ;NEW PSW
066142 005777 113044 TST @SLKCSR ;CHECK FOR KW11-P
066146 013701 001216 MOV $LPVEC,R1 ;KW11-P VECTOR ADDRESS
066152 012721 066262 MOV #CLOCK,(R1)+ ;SET UP KW11-P VECTOR
066156 012711 000300 MOV #300,(R1) ;PSW - PRI 6
066162 012777 177777 113024 MOV #-1,@SLKCSB ;LOAD COUNTER BUFFER WITH 1'S
066170 012777 000135 113014 MOV #135,@SLKCSR ;SET CLOCK - CNT UP, 16MS, CONT INT
066176 000425 BR CKCLK3
066200 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
066204 012737 066242 000004 MOV #CKCLK2,@#ERRVEC ;CHANGE ERROR VECTOR TO CHECK FOR KW11-L
066212 005777 113002 TST @SLKS ;LOOK FOR KW11-L
066216 013701 001222 MOV $LLVEC,R1 ;KW11-L VECTOR ADDRESS
066222 012721 066262 MOV #CLOCK,(R1)+ ;SET UP KW11-L VECTOR
066226 012711 000300 MOV #300,(R1) ;PSW - PRI 6
066232 012777 000100 112760 MOV #100,@SLKS ;SET KW11-L INTERRUPT
066240 000404 BR CKCLK3
066242 062706 000004 CKCLK2: ADD #4,SP ;RESTORE THE STACK POINTER
066246 062716 000002 ADD #2,(SP) ;INCREMENT RETURN, NO CLOCK
066252 012737 000006 000004 CKCLK3: MOV #6,@#ERRVEC ;RESTORE THE ERROR VECTOR
066260 000207 RTS PC
    
```

;ROUTINE TO COUNT CLOCK TICKS

```

066262 062737 000021 001256 CLOCK: ADD #17.,TIME ;ADD 17 MS TO ELAPSED TIME COUNTER
066270 103003 BCC 1$ ;BRANCH IF NO OVERFLOW
066272 012737 177777 001256 MOV #-1,TIME ;OVERFLOW - RESTORE MAXIMUM COUNT
066300 005737 001260 1$: TST WATCH ;IS WATCH ALREADY ZERO ?
066304 001406 BEQ 2$ ;BR IF IT IS
066306 162737 000021 001260 SUB #17.,WATCH ;SUBTRACT 17 MS FROM WATCH DOG COUNTER
066314 100002 BPL 2$ ;BR IF NOT MINUS
066316 005037 001260 CLR WATCH ;CLEAR WATCH DOG COUNTER
066322 000002 2$: RTI ;RETURN
    
```

;ROUTINE TO CALCULATE + AND - 25% TIME TOLERANCE VALUES

```

066324 162706 000004 TOLER: SUB #4,SP ;SETUP STACK
066330 016616 000004 MOV 4(SP),(SP) ;SAVE STACK
066334 013546 MOV @R5+,-(SP) ;GET TIME VALUE
066336 011666 000004 MOV (SP),4(SP) ;MOVE TIME VALUE
066342 011666 000006 MOV (SP),6(SP) ;MOVE VALUE AGAIN
066346 006216 ASR (SP) ;DIVIDE BY 2
066350 006216 ASR (SP) ;DIVIDE BY 2 AGAIN (FOR A TOTAL OF 4)
066352 061666 000004 ADD (SP),4(SP) ;CALCULATE UPPER LIMIT FOR TIMEOUT
066356 162666 000004 SUB (SP)+,4(SP) ;CALCULATE LOWER LIMIT FOR TIMEOUT
066362 000205 RTS R5 ;RETURN WITH TOLERANCES ON THE STACK
    
```

1

.SBTTL SCOPE HANDLER ROUTINE

```

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

```

066364          $SCOPE:
066364 104407          CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
066366 004737 066720  JSR          PC,STOP
066372 032777 040000 112540 1$:    BIT          #BIT14,@SWR          ;;LOOP ON PRESENT TEST?
066400 001402          BEQ          9$          ;;NO IF SW14=0
066402 000137 066702  JMP          $OVER          ;;JUMP OVER SCOPE ROUTINE
066406          9$:
066406 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)
066410 013746 000004          MOV          @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
066414 012737 066434 000004          MOV          #5$,@#ERRVEC          ;;SET FOR TIMEOUT
066422 005737 177060          TST          @#177060          ;;TIME OUT ON XOR?
066426 012637 000004          MOV          (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
066432 000517          BR          $$VLAD          ;;GO TO THE NEXT TEST
066434 022626          5$:    CMP          (SP)+,(SP)+          ;;CLEAR THE STACK AFTER A TIME OUT
066436 012637 000004          MOV          (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
066442 000517          BR          $OVER          ;;LOOP ON THE PRESENT TEST
066444          6$:;#####END OF CODE FOR THE XOR TESTER#####
066444 105737 001103          2$:    TSTB         $ERFLG          ;;HAS AN ERROR OCCURRED?
066450 001465          BEQ          3$          ;;BR IF NO
066452 022737 177777 067270          CMP          #-1,CPSAVE          ;;SEE IF TIMEOUT WAS PREVIOUSLY RECORDED
066460 001455          BEQ          2003$          ;;KICK AROUND ROUTINE IF SO
066462 013746 000004          MOV          ERRVEC,-(SP)          ;;SAVE CONTENTS OF ERROR VECTOR
066466 012737 066504 000004          MOV          #2000$,ERRVEC          ;;SETUP 'TRAP' RETURN ADDRESS
066474 013737 177766 067270          MOV          177766,CPSAVE          ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
066502 000406          BR          2001$
066504 012737 177777 067270 2000$:  MOV          #-1,CPSAVE          ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
066512 012716 066520          MOV          #2001$,(SP)          ;;SETUP RETURN ADDRESS
066516 000002          RTI
066520 012637 000004          2001$:  MOV          (SP)+,ERRVEC          ;;RESTORE CONTENTS OF ERROR VECTOR

066524 022737 177777 067270 2002$:  CMP          #-1,CPSAVE          ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
066532 001430          BEQ          2003$          ;;BRANCH IF SO
066534 032737 000001 067270          BIT          #BIT00,CPSAVE          ;;SEE IF THE POWER MONITOR BIT IS ON
066542 001424          BEQ          2003$          ;;BRANCH TO CONTINUE ROUTINE IF CLEAR
066544 042737 000001 177766          BIC          #BIT00,177766          ;;CLEAR THE BIT FOUND TO BE SET
066552 013746 001140          MOV          SWR,-(SP)          ;;SAVE SWR ADDRESS
066556 017646 000000          MOV          @(SP),-(SP)          ;;SAVE SWR VALUE
066562 012737 000176 001140          MOV          #176,SWR          ;;GET SOFTWARE SWR ADDRESS
066570 011677 112344          MOV          (SP),@SWR          ;;GET CURRENT SWR VALUE
066574 042777 001000 112336          BIC          #BIT09,@SWR          ;;DON'T ALLOW LOOP ON ERROR ON THIS ERROR
066602 104177          EMT          177          ;;CALL SPECIAL POWER FAIL BIT ERROR CALL
066604 012676 000000          MOV          (SP)+,@(SP)          ;;RESTORE SWR TO ORIGINAL VALUE
066610 012637 001140          MOV          (SP)+,SWR          ;;RESTORE SWR ADDRESS
    
```

```

SCOPE HANDLER ROUTINE

066614          2003$:
066614 105037 001103 4$: CLR      $ERFLG      ;;ZERO THE ERROR FLAG
066620 005037 001176 CLR      $TIMES      ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
066624 032777 004000 112306 3$: BIT      #BIT11,@SWR    ;;INHIBIT ITERATIONS?
066632 001011 BNE      1$          ;;BR IF YES
066634 005737 001100 TST      $PASS      ;;IF FIRST PASS OF PROGRAM
066640 001406 BEQ      1$          ;;INHIBIT ITERATIONS
066642 005237 001104 INC      $ICNT      ;;INCREMENT ITERATION COUNT
066646 023737 001176 001104 CMP      $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
066654 002012 BGE      $OVER      ;;BR IF MORE ITERATION REQUIRED
066656 012737 000001 001104 1$: MOV     #1,$ICNT   ;;REINITIALIZE THE ITERATION COUNTER
066664 013737 066716 001176 MOV     $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
066672 105237 001102 $SVLAD: INCB   $STNM   ;;COUNT TEST NUMBERS
066676 011637 001106 MOV     (SP),$LPADR  ;;SAVE SCOPE LOOP ADDRESS
066702 013777 001102 112232 $OVER: MOV   $STNM,@DISPLAY ;;DISPLAY TEST NUMBER
066710 013716 001106 MOV     $LPADR,(SP)  ;;FUDGE RETURN ADDRESS
066714 000002 RTI          ;;FIXES PS
066716 000005 $MXCNT: 5.      ;;MAX. NUMBER OF ITERATIONS

2
3
4
5
66720
66720 012746 000140 STOP: MOV    #PR3,-(SP)  ;;PUT NEW PS ON STACK
66724 012746 066732 MOV    #64$,-(SP)     ;;PUT NEW PC ON STACK
66730 000002 RTI          ;;POP NEW PC AND PS
66732

6
7
8
9
66732 012746 000240 MOV    #PR5,-(SP)     ;;PUT NEW PS ON STACK
66736 012746 066744 MOV    #65$,-(SP)     ;;PUT NEW PC ON STACK
66742 000002 RTI          ;;POP NEW PC AND PS
66744

10 66744 000207 65$: RTS     PC      ;;RETURN

```

1

.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

066746 105037 067272      $ERROR: CLRB      IBSAVE      ;;CLEAR THE ITEM BYTE SAVE LOCATION
066752 104407              CKSWR          ;;TEST FOR CHANGE IN SOFT-SWR
066754 113737 001102 001246      MOVB      $TSTNM,TSTNUM
066762 105237 001103      7$:      INCB      $ERFLG          ;;SET THE ERROR FLAG
066766 001775              BEQ        7$          ;;DON'T LET THE FLAG GO TO ZERO
066770 013777 001102 112144      MOV      $TSTNM,@DISPLAY      ;;DISPLAY TEST NUMBER AND ERROR FLAG
066776 032777 002000 112134      BIT      #BIT10,@SWR          ;;BELL ON ERROR?
067004 001402              BEQ        1$          ;;NO - SKIP
067006 104401 001202              TYPE      $BELL          ;;RING BELL
067012 005237 001112      1$:      INC      $ERTTL          ;;COUNT THE NUMBER OF ERRORS
067016 011637 001116      MOV      (SP), $ERRPC          ;;GET ADDRESS OF ERROR INSTRUCTION
067022 162737 000002 001116      SUB      #2,$ERRPC
067030 117737 112062 001114      MOVB     @ $ERRPC,$ITEMB      ;;STRIP AND SAVE THE ERROR ITEM CODE
067036 032777 001000 112074      BIT      #BIT09,@SWR          ;;SEE IF LOOP ON ERROR IS SET
067044 001060              BNE      1004$          ;;BRANCH AROUND ROUTINE IF SO
067046 122737 000177 001114      CMPB     #177,$ITEMB          ;;SEE IF THIS IS THE POWER FAIL CALL
067054 001454              BEQ      1004$          ;;BRANCH AROUND ROUTINE IF IT IS
067056 105737 067272              TSTB     IBSAVE          ;;SEE IF THIS IS THE 2ND ERROR CALL IN THIS ROUTINE
067062 001047              BNE      1003$          ;;BRANCH IF SO
067064 022737 177777 067270      CMP      #-1,CPSAVE          ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
067072 001445              BEQ      1004$          ;;BRANCH IF SO
067074 013746 000004              MOV      ERRVEC,-(SP)          ;;SAVE CONTENTS OF ERROR VECTOR
067100 012737 067116 000004      MOV      #1000$,ERRVEC          ;;SETUP 'TRAP' RETURN ADDRESS
067106 013737 177766 067270      MOV      177766,CPSAVE          ;;MOVE CPU ERROR REGISTER TO CPSAVE FOR TEST
067114 000406              BR       1001$
067116 012737 177777 067270 1000$:  MOV      #-1,CPSAVE          ;;SET CPU ERROR REGISTER TIMEOUT INDICATOR
067124 012716 067132              MOV      #1001$, (SP)          ;;SETUP RETURN ADDRESS
067130 000002              RTI
067132 012637 000004      1001$:  MOV      (SP)+,ERRVEC          ;;RESTORE CONTENTS OF ERROR VECTOR

067136 022737 177777 067270 1002$:  CMP      #-1,CPSAVE          ;;SEE IF CPSAVE HAS CPU ERR REG TIMEOUT INDICATION
067144 001420              BEQ      1004$          ;;BRANCH IF SO
067146 032737 000001 067270      BIT      #BIT00,CPSAVE          ;;SEE IF POWER MONITOR BIT IS SET IN CPU ERR REG
067154 001414              BEQ      1004$          ;;BRANCH IF OK
067156 042737 000001 177766      BIC      #BIT00,177766          ;;CLEAR THE BIT FOUND SET
067164 113737 001114 067272      MOVB     $ITEMB,IBSAVE          ;;MAKE IBSAVE NON-ZERO FOR DUAL ERROR CALL
067172 112737 000177 001114      MOVB     #177,$ITEMB          ;;SET $ITEMB TO SPECIAL POWER FAIL POINTER
067200 000402              BR       1004$          ;;BRANCH OVER IBSAVE CLEARING

067202 105037 067272      1003$:  CLRB     IBSAVE          ;;CLEAR IBSAVE SO 2ND TIME THROUGH EXITS
067206 032777 020000 111724 1004$:  BIT      #BIT13,@SWR          ;;SKIP TYPEOUT IF SET
067214 001004              BNE      20$          ;;SKIP TYPEOUTS
067216 004737 067274              JSR      PC,$ERRTYP          ;;GO TO USER ERROR ROUTINE
    
```



```

067222 104401 001207          TYPE      ,SCLRF
067226          20$:
067226 105737 067272          2$:      TSTB      IBSAVE      ;;SEE IF IBSAVE IS LOADED
067232 001005          BNE        3$          ;;BRANCH IF NOT - NO HALT ON PWR MON BIT ERROR
067234 005777 111700          TST        @SWR      ;;HALT ON ERROR
067240 100002          BPL        3$          ;;SKIP IF CONTINUE
067242 000000          HALT          ;;HALT ON ERROR!
067244 104407          CKSWR      ;;TEST FOR CHANGE IN SOFT-SWR
067246          3$:
067246 022737 066110 000042    CMP        #SCLNDAD,@#42  ;;ACT-11 AUTO-ACCEPT?
067254 001001          BNE        6$          ;;BRANCH IF NO
067256 000000          HALT          ;;YES
067260          6$:
067260 105737 067272          TSTB      IBSAVE      ;;SEE IF ITEM BYTE SAVE LOCATION HAS AN ERROR CALL
067264 001236          BNE        7$          ;;BRANCH BACK TO CALL ORIGIN'L ERROR
067266 000002          RTI          ;;RETURN
067270 000000          CPSAVE: .WORD 0      ;;LOCATION TO SAVE CPU ERROR REG CONTENTS
067272 000000          IBSAVE: .WORD 0      ;;LOCATION TO SAVE ITEM BYTE

```

.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.

067274			\$ERRTYP:		
067274	104401	001207		TYPE	,\$SCLRF
067300	010046			MOV	RO,-(SP)
067302	005000			CLR	RO
067304	153700	001114		BISB	@#\$ITEMB,RO
067310	001004			BNE	1\$
					::'CARRIAGE RETURN' & 'LINE FEED'
					::SAVE RO
					::PICKUP THE ITEM INDEX
067312	013746	001116		MOV	\$ERRPC,-(SP)
					::IF ITEM NUMBER IS ZERO, JUST
					::TYPE THE PC OF THE ERROR
					::SAVE \$ERRPC FOR TYPEOUT
					::ERROR ADDRESS
067316	104402			TYPOC	
067320	000456			BR	10\$
067322	122700	000177	1\$:	CMPB	#177,RO
067326	001006			BNE	1000\$
067330	113737	001102	067632	MOVB	\$STNM,PFTSTN
067336	012700	067472		MOV	#PFECB,RO
067342	000406			BR	1001\$
067344	005300		1000\$:	DEC	RO
067346	006300			ASL	RO
067350	006300			ASL	RO
067352	006300			ASL	RO
067354	062700	001310		ADD	#\$ERRTB,RO
067360	012037	067370	1001\$:	MOV	(RO)+,2\$
067364	001404			BEQ	3\$
067366	104401			TYPE	
067370	000000		2\$:	.WORD	0
067372	104401	001207		TYPE	,\$SCLRF
067376	012037	067406	3\$:	MOV	(RO)+,4\$
067402	001404			BEQ	5\$
067404	104401			TYPE	
067406	000000		4\$:	.WORD	0
067410	104401	001207		TYPE	,\$SCLRF
067414	010146		5\$:	MOV	R1,-(SP)
067416	012001			MOV	(R0)+,R1
067420	001415			BEQ	9\$
067422	012000			MOV	(R0)+,RO
067424	105720		6\$:	TSTB	(R0)+
067426	001003			BNE	7\$
067430	013146			MOV	@(R1)+,-(SP)
067432	104402			TYPOC	
067434	000402			BR	8\$
067436			7\$:		
067436	013146			MOV	@(R1)+,-(SP)
067440	104405			TYPDS	
067442	005711		8\$:	TST	(R1)
067444	001403			BEQ	9\$
067446	104401	067466		TYPE	.11\$
067452	000764			BR	6\$
					::SAVE @(R1)+ FOR TYPEOUT
					::GO TYPE--DECIMAL ASCII WITH SIGN
					::IS THERE ANOTHER NUMBER?
					::BR IF NO
					::TYPE TWO(2) SPACES
					::LOOP
067454	012601		9\$:	MOV	(SP)+,R1
067456	012600		10\$:	MOV	(SP)+,RO
					::RESTORE R1
					::RESTORE RO

067460	104401	001207			TYPE	,\$CRLF	:::'CARRIAGE RETURN' & 'LINE FEED'
067464	000207				RTS	PC	:::RETURN
067466	040	040	000	11\$:	.ASCIZ	/ /	:::TWO(2) SPACES
					.EVEN		
067472	067502	067564	067616	PFECH:	PFECH1,PFECH2,PFECH3,PFECH4	:::WORDS DEFINING TABLES BFLOW	
067502	120	117	127	PFECH1:	.ASCIZ	?POWER MONITOR BIT IN CPU ERROR REGISTER FOUND SET?	
067564	124	105	123	PFECH2:	.ASCIZ	?TESTNO ERR PC CPUERREG?	
					.EVEN		
067616	067632	001116	067270	PFECH3:	.WORD	PFTSTN,\$ERRPC,CPSAVE,0	
067626	000	000	000	PFECH4:	.BYTE	0,0,0,0	
067632	000000			PFTSTN:	.WORD	0	:::CONTAINS TEST NUMBER FOR PF BIT ERROR

.SBTTL TYPE ROUTINE

```

*****
*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

```

```

067634 105737 001157 $TYPE: TSTB $TFPLG      ;; IS THERE A TERMINAL?
067640 100002      BPL 1$      ;; BR IF YES
067642 000000      HALT      ;; HALT HERE IF NO TERMINAL
067644 000407      BR 3$      ;; LEAVE
067646 010046 1$: MOV RO,-(SP)  ;; SAVE RO
067650 017600 000002 MOV @2(SP),RO  ;; GET ADDRESS OF ASCIZ STRING
067654 112046 2$: MOVB (RO)+,-(SP)  ;; PUSH CHARACTER TO BE TYPED ONTO STACK
067656 001005      BNE 4$      ;; BR IF IT ISN'T THE TERMINATOR
067660 005726      TST (SP)+  ;; IF TERMINATOR POP IT OFF THE STACK
067662 012600 60$: MOV (SP)+,RO  ;; RESTORE RO
067664 062716 000002 3$: ADD #2,(SP)  ;; ADJUST RETURN PC
067670 000002      RTI      ;; RETURN
067672 122716 000011 4$: CMPB #HT,(SP)  ;; BRANCH IF <HT>
067676 001430      BEQ 8$      ;;
067700 122716 000200      CMPB #CRLF,(SP)  ;; BRANCH IF NOT <CRLF>
067704 001006      BNE 5$      ;;
067706 005726      TST (SP)+  ;; POP <CR><LF> EQUIV
067710 104401      TYPE      ;; TYPE A CR AND LF
067712 001207      $CRLF
067714 105037 070122 CLRB $CHARCNT  ;; CLEAR CHARACTER COUNT
067720 000755      BR 2$      ;; GET NEXT CHARACTER
067722 004737 070004 5$: JSR PC,$TYPEC  ;; GO TYPE THIS CHARACTER
067726 123726 001156 6$: CMPB $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
067732 001350      BNE 2$      ;; IF NO GO GET NEXT CHAR.
067734 013746 001154      MOV $NULL,-(SP)  ;; GET # OF FILLER CHARS. NEEDED
                                ;; AND THE NULL CHAR.
067740 105366 000001 7$: DECB 1(SP)  ;; DOES A NULL NEED TO BE TYPED?
067744 002770      BLT 6$      ;; BR IF NO--GO POP THE NULL OFF OF STACK
067746 004737 070004      JSR PC,$TYPEC  ;; GO TYPE A NULL
067752 105337 070122      DECB $CHARCNT  ;; DO NOT COUNT AS A COUNT
067756 000770      BR 7$      ;; LOOP

```

;HORIZONTAL TAB PROCESSOR

```

067760 112716 000040 8$: MOVB #' ,(SP)  ;; REPLACE TAB WITH SPACE
067764 004737 070004 9$: JSR PC,$TYPEC  ;; TYPE A SPACE
067770 132737 000007 070122 BITB #7,$CHARCNT  ;; BRANCH IF NOT AT
067776 001372      BNE 9$      ;; TAB STOP
070000 005726      TST (SP)+  ;; POP SPACE OFF STACK
070002 000724      BR 2$      ;; GET NEXT CHARACTER

```

```

070004                                $TYPEC:
070004 105777 111134                   TSTB   @STKS   ;;CHAR IN KYBD BUFFER?
070010 100022                           BPL     10$    ;;BR IF NOT
070012 017746 111130                   MOV     @STKB,-(SP) ;;GET CHAR
070016 042716 177600                   BIC     #177600,(SP) ;;STRIP EXTRANEIOUS BITS
070022 122716 000023                   CMPB   #SXOFF,(SP) ;;WAS CHAR XOFF
070026 001012                           BNE     102$   ;;BR IF NOT
070030                                101$:
070030 105777 111110                   TSTB   @STKS   ;;WAIT FOR CHAR
070034 100375                           BPL     101$
070036 117716 111104                   MOVB   @STKB,(SP) ;;GET CHAR
070042 042716 177600                   BIC     #177600,(SP) ;;STRIP IT
070046 122716 000021                   CMPB   #SXON,(SP) ;;WAS IT XON?
070052 001366                           BNE     101$   ;;BR IF NOT
070054                                102$:
070054 005726                           TST     (SP)+   ;;FIX STACK
070056                                10$:
070056 105777 111066                   TSTB   @STPS   ;;WAIT UNTIL PRINTER IS READY
070062 100375                           BPL     10$
070064 116677 000002 111060             MOVB   2(SP),@STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
070072 122766 000015 000002             CMPB   #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
070100 001003                           BNE     1$     ;;BRANCH IF NO
070102 105037 070122                   CLRB   $CHARCNT ;;YES--CLEAR CHARACTER COUNT
070106 000406                           BR     $TYPEX ;;EXIT
070110 122766 000012 000002 1$:        CMPB   #LF,2(SP) ;;IS CHARACTER A LINE FEED?
070116 001402                           BEQ    $TYPEX ;;BRANCH IF YES
070120 105227                           INCB   (PC)+   ;;COUNT THE CHARACTER
070122 000000                           $CHARCNT: WORD 0 ;;CHARACTER COUNT STORAGE
070124 000207                           $TYPEX: RTS   PC

```

.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
  
```

070126	017646	000000		\$TYPOS:	MOV	@(SP),-(SP)	;;PICKUP THE MODE
070132	116637	000001	070351		MOVB	1(SP), \$OFILL	;;LOAD ZERO FILL SWITCH
070140	112637	070353			MOVB	(SP)+, \$OMODE+1	;;NUMBER OF DIGITS TO TYPE
070144	062716	000002			ADD	#2,(SP)	;;ADJUST RETURN ADDRESS
070150	000406				BR	\$TYPON	
070152	112737	000001	070351	\$TYPOC:	MOVB	#1,\$OFILL	;;SET THE ZERO FILL SWITCH
070160	112737	000006	070353		MOVB	#6,\$OMODE+1	;;SET FOR SIX(6) DIGITS
070166	112737	000005	070350	\$TYPON:	MOVB	#5,\$OCNT	;;SET THE ITERATION COUNT
070174	010346				MOV	R3,-(SP)	;;SAVE R3
070176	010446				MOV	R4,-(SP)	;;SAVE R4
070200	010546				MOV	R5,-(SP)	;;SAVE R5
070202	113704	070353			MOVB	\$OMODE+1,R4	;;GET THE NUMBER OF DIGITS TO TYPE
070206	005404				NEG	R4	
070210	062704	000006			ADD	#6,R4	;;SUBTRACT IT FOR MAX. ALLOWED
070214	110437	070352			MOVB	R4,\$OMODE	;;SAVE IT FOR USE
070220	113704	070351			MOVB	\$OFILL,R4	;;GET THE ZERO FILL SWITCH
070224	016605	000012			MOV	12(SP),R5	;;PICKUP THE INPUT NUMBER
070230	005003				CLR	R3	;;CLEAR THE OUTPUT WORD
070232	006105			1\$:	ROL	R5	;;ROTATE MSB INTO 'C'
070234	000404				BR	3\$;;GO DO MSB
070236	006105			2\$:	ROL	R5	;;FORM THIS DIGIT
070240	006105				ROL	R5	
070242	006105				ROL	R5	
070244	010503				MOV	R5,R3	
070246	006103			3\$:	ROL	R3	;;GET LSB OF THIS DIGIT
070250	105337	070352			DECB	\$OMODE	;;TYPE THIS DIGIT?
070254	100016				BPL	7\$;;BR IF NO
070256	042703	177770			BIC	#177770,R3	;;GET RID OF JUNK
070262	001002				BNE	4\$;;TEST FOR 0
070264	005704				TST	R4	;;SUPPRESS THIS 0?
070266	001403				BEQ	5\$;;BR IF YES
070270	005204			4\$:	INC	R4	;;DON'T SUPPRESS ANYMORE 0'S

```

070272 052703 000060          BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
070276 052703 000040          5$:    BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
070302 110337 070346          MOVB    R3,8$      ;;SAVE FOR TYPING
070306 104401 070346          TYPE   ,8$      ;;GO TYPE THIS DIGIT
070312 105337 070350          7$:    DECB   $OCNT  ;;COUNT BY 1
070316 003347          BGT    2$      ;;BR IF MORE TO DO
070320 002402          BLT    6$      ;;BR IF DONE
070322 005204          INC    R4      ;;INSURE LAST DIGIT ISN'T A BLANK
070324 000744          BR     2$      ;;GO DO THE LAST DIGIT
070326 012605          6$:    MOV    (SP)+,R5  ;;RESTORE R5
070330 012604          MOV    (SP)+,R4  ;;RESTORE R4
070332 012603          MOV    (SP)+,R3  ;;RESTORE R3
070334 016666 000002 000004  MOV    2(SP),4(SP) ;;SET THE STACK FOR RETURNING
070342 012616          MOV    (SP)+,(SP)
070344 000002          RTI      ;;RETURN
070346 000          8$:    .BYTE 0      ;;STORAGE FOR ASCII DIGIT
070347 000          .BYTE 0      ;;TERMINATOR FOR TYPE ROUTINE
070350 000          $OCNT: .BYTE 0  ;;OCTAL DIGIT COUNTER
070351 000          $OFILL: .BYTE 0 ;;ZERO FILL SWITCH
070352 000000          $OMODE: .WORD 0 ;;NUMBER OF DIGITS TO TYPE
    
```

.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

070354				\$TYPDS:	MOV R0,-(SP)	:::PUSH R0 ON STACK
070354	010046				MOV R1,-(SP)	:::PUSH R1 ON STACK
070356	010146				MOV R2,-(SP)	:::PUSH R2 ON STACK
070360	010246				MOV R3,-(SP)	:::PUSH R3 ON STACK
070362	010346				MOV R5,-(SP)	:::PUSH R5 ON STACK
070364	010546				MOV #20200,-(SP)	:::SET BLANK SWITCH AND SIGN
070366	012746	020200			MOV 20(SP),R5	:::GET THE INPUT NUMBER
070372	016605	000020			BPL 1\$:::BR IF INPUT IS POS.
070376	100004				NEG R5	:::MAKE THE BINARY NUMBER POS.
070400	005405				MOVB #'-,1(SP)	:::MAKE THE ASCII NUMBER NEG.
070402	112766	000055	000001	1\$:	CLR R0	:::ZERO THE CONSTANTS INDEX
070410	005000				MOV #SDBLK,R3	:::SETUP THE OUTPUT POINTER
070412	012703	070570			MOVB #' ,(R3)+	:::SET THE FIRST CHARACTER TO A BLANK
070416	112723	000040		2\$:	CLR R2	:::CLEAR THE BCD NUMBER
070422	005002				MOV \$DTBL(R0),R1	:::GET THE CONSTANT
070424	016001	070560		3\$:	SUB R1,R5	:::FORM THIS BCD DIGIT
070430	160105				BLT 4\$:::BR IF DONE
070432	002402				INC R2	:::INCREASE THE BCD DIGIT BY 1
070434	005202				BR 3\$	
070436	000774				ADD R1,R5	:::ADD BACK THE CONSTANT
070440	060105			4\$:	TST R2	:::CHECK IF BCD DIGIT=0
070442	005702				BNE 5\$:::FALL THROUGH IF 0
070444	001002				TSTB (SP)	:::STILL DOING LEADING 0'S?
070446	105716				BMI 7\$:::BR IF YES
070450	100407				ASLB (SP)	:::MSD?
070452	106316			5\$:	BCC 6\$:::BR IF NO
070454	103003				MOVB 1(SP),-1(R3)	:::YES--SET THE SIGN
070456	116663	000001	177777	6\$:	BIS #'0,R2	:::MAKE THE BCD DIGIT ASCII
070464	052702	000060		7\$:	BIS #' ,R2	:::MAKE IT A SPACE IF NOT ALREADY A DIGIT
070470	052702	000040			MOVB R2,(R3)+	:::PUT THIS CHARACTER IN THE OUTPUT BUFFER
070474	110223				TST (R0)+	:::JUST INCREMENTING
070476	005720				CMR R0,#10	:::CHECK THE TABLE INDEX
070500	020027	000010			BLT 2\$:::GO DO THE NEXT DIGIT
070504	002746				BGT 8\$:::GO TO EXIT
070510	010502				MOV R5,R2	:::GET THE LSD
070512	000764				BR 6\$:::GO CHANGE TO ASCII
070514	105726			8\$:	TSTB (SP)+	:::WAS THE LSD THE FIRST NON-ZERO?
070516	100003				BPL 9\$:::BR IF NO
070520	116663	177777	177776		MOVB -1(SP),-2(R3)	:::YES--SET THE SIGN FOR TYPING
070526	105013			9\$:	CLRB (R3)	:::SET THE TERMINATOR
070530	012605				MOV (SP)+,R5	:::POP STACK INTO R5
070532	012603				MOV (SP)+,R3	:::POP STACK INTO R3
070534	012602				MOV (SP)+,R2	:::POP STACK INTO R2
070536	012601				MOV (SP)+,R1	:::POP STACK INTO R1


```
070540 012600          MOV      (SP)+,RO      ;;POP STACK INTO RO
070542 104401 070570  TYPE      ,SDBLK      ;;NOW TYPE THE NUMBER
070546 016666 000002 000004  MOV      2(SP),4(SP)  ;;ADJUST THE STACK
070554 012616          MOV      (SP)+,(SP)
070556 000002          RTI                          ;;RETURN TO USER
070560 023420          $DTBL: 10000.
070562 001750          1000.
070564 000144          100.
070566 000012          10.
070570          $SDBLK: .BLKW 4
```

.SBTTL TTY INPUT ROUTINE

```

*****
ENABL LSB
070600 000000 $TKCNT: .WORD 0 ;;NUMBER OF ITEMS IN QUEUE
070602 000000 $TKQIN: .WORD 0 ;;INPUT POINTER
070604 000000 $TKQOUT: .WORD 0 ;;OUTPUT POINTER
070606 070607 $TKQSRT: .BLKB 1 ;;TTY KEYBOARD QUEUE
$TKQEND=.
.EVEN

;*TK INITIALIZE ROUTINE
;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
:
;*CALL:
:
:   JSR   PC,$TKINT
:   RETURN
:
070610 005037 070600 $TKINT: CLR   $TKCNT ;;CLEAR COUNT OF ITEMS IN QUEUE
070614 012737 070606 070602   MOV   #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
070622 013737 070602 070604   MOV   $TKQIN,$TKQOUT ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
070630 012737 070660 000060   MOV   #$TKSRV,@TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
070636 012737 000200 000062   MOV   #200,@TKVEC+2 ;;'BR' LEVEL 4
070644 005777 110276   TST   @TKB ;;CLEAR DONE FLAG
070650 012777 000100 110266   MOV   #100,@TKS ;;ENABLE TTY KEYBOARD INTERRUPT
070656 000207   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.
;*IF THE CHARACTER IS A 'CONTROL-C' (^C) $TKINT IS CALLED AND
;*UPON RETURN EXIT IS MADE TO THE 'CONTROL-C' RESTART ADDRESS (START)
:
$TKSRV: MOVB  @TKB,-(SP) ;;PICKUP THE CHARACTER
        BIC   #^C177,(SP) ;;STRIP THE JUNK
        CMP   (SP),#$XON ;;IS IT A RANDOM XON?
        BNE   30$ ;;BRANCH IF NO
        TST  (SP)+ ;;CLEAN RANDOM XON OFF STACK
        RTI   ;;RETURN
30$:
        CMP   (SP),#3 ;;IS IT A CONTROL C?
        BNE   1$ ;;BRANCH IF NO
        TYPE  ,SCNTLC ;;TYPE A CONTROL-C (^C)
        JSR   PC,$TKINT ;;INIT THE KEYBOARD
        TST  (SP)+ ;;CLEAN UP STACK
        JMP   START ;;CONTROL C RESTART
1$:
        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$:
070744 022737 000001 070600   CMP   #1,$TKCNT ;;IS THE QUEUE FULL?
070752 001004   BNE   3$ ;;BRANCH IF NO
070754 104401 001202   TYPE  ,SBELL ;;RING THE TTY BELL

```

```

070760 005726          TST      (SP)+      ;; CLEAN CHARACTER OFF OF STACK
070762 000451          BR        5$          ;; EXIT
070764 021627 000023  3$:      CMP      (SP),#23      ;; IS IT A CONTROL-S?
070770 001021          BNE      32$          ;; BRANCH IF NO
070772 005077 110146          CLR      @STKS        ;; DISABLE TTY KEYBOARD INTERRUPTS
070776 005726          TST      (SP)+      ;; CLEAN CHAR OFF STACK
071000 105777 110140  31$:      TSTB     @STKS        ;; WAIT FOR A CHAR
071004 100375          BPL      31$          ;; LOOP UNTIL ITS THERE
071006 117746 110134          MOVB     @STKB,-(SP)  ;; GET THE CHARACTER
071012 042716 177600          BIC      #^C177,(SP) ;; MAKE IT 7-BIT ASCII
071016 022627 000021          CMP      (SP)+,#21  ;; IS IT A CONTROL-Q?
071022 001366          BNE      31$          ;; BRANCH IF NO
071024 012777 000100 110112          MOV      #100,@STKS ;; REENABLE TTY KEYBOARD INTERRUPTS
071032 000002          RTI          ;; RETURN
071034 005237 070600  32$:      INC      $TKCNT      ;; COUNT THIS CHARACTER
071040 021627 000140          CMP      (SP),#140  ;; IS IT UPPER CASE?
071044 002405          BLT      4$          ;; BRANCH IF YES
071046 021627 000175          CMP      (SP),#175  ;; IS IT A SPECIAL CHAR?
071052 003002          BGT      4$          ;; BRANCH IF YES
071054 042716 000040          BIC      #40,(SP)   ;; MAKE IT UPPER CASE
071060 112677 177516  4$:      MOVB     (SP)+,@STKQIN ;; AND PUT IT IN QUEUE
071064 005237 070602          INC      $TKQIN      ;; UPDATE THE POINTER
071070 023727 070602 070607          CMP      $TKQIN,#$TKQEND ;; GO OFF THE END?
071076 001003          BNE      5$          ;; BRANCH IF NO
071100 012737 070606 070602          MOV      #$TKQSRT,$TKQIN ;; RESET THE POINTER
071106 000002  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.

```

071110 022737 000176 001140 $CKSWR: CMP      #SWREG,SWR      ;; IS THE SOFT-SWR SELECTED
071116 001124          BNE      15$          ;; EXIT IF NOT
071120 105777 110020          TSTB     @STKS        ;; IS A CHAR WAITING?
071124 100121          BPL      15$          ;; IF NOT, EXIT
071126 117746 110014          MOVB     @STKB,-(SP)  ;; YES
071132 042716 177600          BIC      #^C177,(SP)  ;; MAKE IT 7-BIT ASCII
071136 021627 000007          CMP      (SP),#7     ;; IS IT A CONTROL-G?
071142 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.

```

071144 123727 001134 000001 6$:      CMPB     $AUTOB,#1    ;; ARE WE RUNNING IN AUTO-MODE?
071152 001674          BEQ      2$          ;; BRANCH IF YES
071154 005726          TST      (SP)+      ;; CLEAR CONTROL-G OFF STACK
071156 004737 070610          JSR      PC,$TKINT   ;; FLUSH THE TTY INPUT QUEUE
071162 005077 107756          CLR      @STKS        ;; DISABLE TTY KEYBOARD INTERRUPTS
071166 112737 000001 001135          MOVB     #1,$INTAG    ;; SET INTERRUPT MODE INDICATOR

071174 104401 072020          TYPE     ,SCNTLG      ;; ECHO THE CONTROL-G (^G)
071200 104401 072025          $GTSWR: TYPE     ,SMSWR      ;; TYPE CURRENT CONTENTS
071204 013746 000176          MOV      SWREG,-(SP)  ;; SAVE SWREG FOR TYPEOUT
071210 104402          TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)

```

```

071212 104401 072036          TYPE      ,SMNEW          ;;PROMPT FOR NEW SWR
071216 005046          CLR      -(SP)          ;;CLEAR COUNTER
071220 005046          CLR      -(SP)          ;;THE NEW SWR
071222 105777 107716      7$:      TSTB     @$TKS          ;;CHAR THERE?
071226 100375          BPL      7$             ;;IF NOT TRY AGAIN

071230 117746 107712      MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
071234 042716 177600      BIC     #^C177,(SP)    ;;MAKE IT 7-BIT ASCII

071240 021627 000003      CMP     (SP),#3        ;;IS IT A CONTROL-C?
071244 001015          BNE     9$             ;;BRANCH IF NOT
071246 104401 072006      TYPE    ,SCNTLC       ;;YES, ECHO CONTROL-C (^C)
071252 062706 000006      ADD     #6,SP          ;;CLEAN UP STACK
071256 123727 001135 000001  CMPB    $INTAG,#1      ;;REENABLE TTY KEYBOARD INTERRUPTS?
071264 001003          BNE     8$             ;;BRANCH IF NO
071266 012777 000100 107650  MOV     #100,@$TKS     ;;ALLOW TTY KEYBOARD INTERRUPTS
071274 000137 002240      8$:      JMP     START          ;;CONTROL-C RESTART

071300 021627 000025      9$:      CMP     (SP),#25     ;;IS IT A CONTROL-U?
071304 001005          BNE     10$          ;;BRANCH IF NOT
071306 104401 072013      TYPE    ,SCNTLU       ;;YES, ECHO CONTROL-U (^U)
071312 062706 000006      20$:     ADD     #6,SP          ;;IGNORE PREVIOUS INPUT
071316 000737          BR      19$          ;;LET'S TRY IT AGAIN

071320 021627 000015      10$:     CMP     (SP),#15     ;;IS IT A <CR>?
071324 001022          BNE     16$          ;;BRANCH IF NO
071326 005766 000004      TST     4(SP)         ;;YES, IS IT THE FIRST CHAR?
071332 001403          BEQ    11$          ;;BRANCH IF YES
071334 016677 000002 107576  MOV     2(SP),@SWR     ;;SAVE NEW SWR
071342 062706 000006      11$:     ADD     #6,SP          ;;CLEAR UP STACK
071346 104401 001207      14$:     TYPE    ,SCRLF       ;;ECHO <CR> AND <LF>
071352 123727 001135 000001  CMPB    $INTAG,#1      ;;RE-ENABLE TTY KBD INTERRUPTS?
071360 001003          BNE     15$          ;;BRANCH IF NOT
071362 012777 000100 107554  MOV     #100,@$TKS     ;;RE-ENABLE TTY KBD INTERRUPTS
071370 000002          RTI                    ;;RETURN
071372 004737 070004      15$:     JSR     PC,$TYPEC    ;;ECHO CHAR
071376 021627 000060      16$:     CMP     (SP),#60     ;;CHAR < 0?
071402 002420          BLT    18$          ;;BRANCH IF YES
071404 021627 000067      CMP     (SP),#67     ;;CHAR > 7?
071410 003015          BGT    18$          ;;BRANCH IF YES
071412 042726 000060      BIC     #60,(SP)+     ;;STRIP-OFF ASCII
071416 005766 000002      TST     2(SP)         ;;IS THIS THE FIRST CHAR
071422 001403          BEQ    17$          ;;BRANCH IF YES
071424 006316          ASL    (SP)          ;;NO, SHIFT PRESENT
071426 006316          ASL    (SP)          ;;CHAR OVER TO MAKE
071430 006316          ASL    (SP)          ;;ROOM FOR NEW ONE.
071432 005266 000002      17$:     INC     2(SP)         ;;KEEP COUNT OF CHAR
071436 056616 177776      BIS     -2(SP),(SP)   ;;SET IN NEW CHAR
071442 000667          BR      7$           ;;GET THE NEXT ONE
071444 104401 001206      18$:     TYPE    ,SQUES     ;;TYPE ?<CR><LF>
071450 000720          BR      20$          ;;SIMULATE CONTROL-U
.DSABL  LSB

```

;;*****

```

: * THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
: * CALL:
: * RDCHR RETURN HERE :: GET A CHARACTER FROM THE QUEUE
: * :: CHARACTER IS ON THE STACK
: * :: WITH PARITY BIT STRIPPED OFF
:
071452 011646 SRDCHR: MOV (SP),-(SP) :: PUSH DOWN THE PC AND
071454 016666 000004 000002 MOV 4(SP),2(SP) :: THE PS
071462 005066 000004 CLR 4(SP) :: GET READY FOR A CHARACTER
071466 005046 CLR -(SP) :: PUT NEW PS ON STACK
071470 012746 071476 MOV #64$,-(SP) :: PUT NEW PC ON STACK
071474 000002 RTI :: POP NEW PC AND PS
071476
071476 005737 070600 64$: TST $TKCNT :: WAIT ON A CHARACTER
071502 001775 1$: BEQ 1$
071504 005337 070600 DEC $TKCNT :: DECREMENT THE COUNTER
071510 117766 177070 000004 MOV @STKQOUT,4(SP) :: GET ONE CHARACTER
071516 005237 070604 INC $TKQOUT :: UPDATE THE POINTER
071522 023727 070604 070607 CMP $TKQOUT,#$TKQEND :: DID IT GO OFF OF THE END?
071530 001003 BNE 2$ :: BRANCH IF NO
071532 012737 070606 070604 MOV #$TKQSRT,$TKQOUT :: RESET THE POINTER
071540 000002 RTI :: RETURN
: *****
: * THIS ROUTINE WILL INPUT A STRING FROM THE TTY
: * CALL:
: * RDLIN RETURN HERE :: INPUT A STRING FROM THE TTY
: * :: ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
: * :: TERMINATOR WILL BE A BYTE OF ALL 0'S
:
071542 010346 SRDLIN: MOV R3, -(SP) :: SAVE R3
071544 005046 CLR -(SP) :: CLEAR THE RUBOUT KEY
071546 012703 071776 1$: MOV #$TTYIN,R3 :: GET ADDRESS
071552 022703 072006 2$: CMP #$TTYIN+8.,R3 :: BUFFER FULL?
071556 101456 BLOS 4$ :: BR IF YES
071560 104410 RDCHR :: GO READ ONE CHARACTER FROM THE TTY
071562 112613 MOV (SP)+,(R3) :: GET CHARACTER
071564 122713 000177 10$: CMPB #177,(R3) :: IS IT A RUBOUT
071570 001022 BNE 5$ :: BR IF NO
071572 005716 TST (SP) :: IS THIS THE FIRST RUBOUT?
071574 001007 BNE 6$ :: BR IF NO
071576 112737 000134 071774 MOVB #' \ ,9$ :: TYPE A BACK SLASH
071604 104401 071774 TYPE ,9$
071610 012716 177777 MOV #-1,(SP) :: SET THE RUBOUT KEY
071614 005303 6$: DEC R3 :: BACKUP BY ONE
071616 020327 071776 CMP R3,$$TTYIN :: STACK EMPTY?
071622 103434 BLO 4$ :: BR IF YES
071624 111337 071774 MOV (R3),9$ :: SETUP TO TYPEOUT THE DELETED CHAR.
071630 104401 071774 TYPE ,9$ :: GO TYPE
071634 000746 BR 2$ :: GO READ ANOTHER CHAR.
071636 005716 5$: TST (SP) :: RUBOUT KEY SET?
071640 001496 BEQ 7$ :: BR IF NO
071642 112737 000134 071774 MOVB #' \ ,9$ :: TYPE A BACK SLASH
071650 104401 071774 TYPE ,9$
071654 005016 CLR (SP) :: CLEAR THE RUBOUT KEY
071656 122713 000025 7$: CMPB #25,(R3) :: IS CHARACTER A CTRL U?
071662 001003 BNE 8$ :: BR IF NO
  
```

```

071664 104401 072013          TYPE ,SCNTLU          ;;TYPE A CONTROL 'U'
071670 000726          BR 1$          ;;GO START OVER
071672 122713 000022      8$: CMPB #22,(R3)      ;;IS CHARACTER A '^R'?
071676 001011          BNE 3$          ;;BRANCH IF NO
071700 105013          CLRB (R3)          ;;CLEAR THE CHARACTER
071702 104401 001207      TYPE ,SCRLF          ;;TYPE A 'CR' & 'LF'
071706 104401 071776      TYPE ,STTYIN          ;;TYPE THE INPUT STRING
071712 000717          BR 2$          ;;GO PICKUP ANOTHER CHACTER
071714 104401 001206      4$: TYPE ,SQUES          ;;TYPE A '?'
071720 000712          BR 1$          ;;CLEAR THE BUFFER AND LOOP
071722 111337 071774      3$: MOVB (R3),9$          ;;ECHO THE CHARACTER
071726 104401 071774      TYPE ,9$
071732 122723 000015      CMPB #15,(R3)+          ;;CHECK FOR RETURN
071736 001305          BNE 2$          ;;LOOP IF NOT RETURN
071740 105063 177777      CLRB -1(R3)          ;;CLEAR RETURN (THE 15)
071744 104401 001210      TYPE ,SLF          ;;TYPE A LINE FEED
071750 005726          TST (SP)+          ;;CLEAN RUBOUT KEY FROM THE STACK
071752 012603          MOV (SP)+,R3          ;;RESTORE R3
071754 011646          MOV (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
071756 016666 000004 000002 MOV 4(SP),2(SP)          ;; FIRST ASCII CHARACTER ON IT
071764 012766 071776 000004 MOV #STTYIN,4(SP)
071772 000002          RTI          ;;RETURN
071774 000          9$: .BYTE 0          ;;STORAGE FOR ASCII CHAR. TO TYPE
071775 000          .BYTE 0          ;;TERMINATOR
071776          STTYIN: .BLKB 8          ;;RESERVE 8 BYTES FOR TTY INPUT
072006 136 103 015 SCNTLC: .ASCIZ /^C/<15><12>          ;;CONTROL 'C'
072013 136 125 015 SCNTLU: .ASCIZ /^U/<15><12>          ;;CONTROL 'U'
072020 136 107 015 SCNTLG: .ASCIZ /^G/<15><12>          ;;CONTROL 'G'
072025 015 012 123 SMSWR: .ASCIZ <15><12>/SWR = /
072036 040 040 116 SMNEW: .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
  
```

```

072050 011646          $RDOCT: MOV      (SP),-(SP)      ;;PROVIDE SPACE FOR THE
072052 016666 000004 000002 MOV      4(SP),2(SP)      ;;INPUT NUMBER
072060 010046          MOV      R0,-(SP)          ;;PUSH R0 ON STACK
072062 010146          MOV      R1,-(SP)          ;;PUSH R1 ON STACK
072064 010246          MOV      R2,-(SP)          ;;PUSH R2 ON STACK
072066 104411          1$:  RDLIN                    ;;READ AN ASCII LINE
072070 012600          MOV      (SP)+,R0          ;;GET ADDRESS OF 1ST CHARACTER
072072 010037 072176  MOV      R0,5$             ;;AND SAVE IT
072076 005001          CLR      R1              ;;CLEAR DATA WORD
072100 005002          CLR      R2
072102 112046          2$:  MOVB     (R0)+,-(SP)      ;;PICKUP THIS CHARACTER
072104 001420          BEQ      3$              ;;IF ZERO GET OUT
072106 122716 000060  CMPB     #'0,(SP)          ;;MAKE SURE THIS CHARACTER
072112 003026          BGT      4$              ;;IS AN OCTAL DIGIT
072114 122716 000067  CMPB     #'7,(SP)
072120 002423          BLT      4$
072122 006301          ASL     R1              ;;*2
072124 006102          ROL     R2
072126 006301          ASL     R1              ;;*4
072130 006102          ROL     R2
072132 006301          ASL     R1              ;;*8
072134 006102          ROL     R2
072136 042716 177770  BIC     #'C7,(SP)          ;;STRIP THE ASCII JUNK
072142 062601          ADD     (SP)+,R1          ;;ADD IN THIS DIGIT
072144 000756          BR      2$              ;;LOOP
072146 005726          3$:  TST     (SP)+          ;;CLEAN TERMINATOR FROM STACK
072150 010166 000012  MOV     R1,12(SP)        ;;SAVE THE RESULT
072154 010237 072206  MOV     R2,$HIOCT
072160 012602          MOV     (SP)+,R2          ;;POP STACK INTO R2
072162 012601          MOV     (SP)+,R1          ;;POP STACK INTO R1
072164 012600          MOV     (SP)+,R0          ;;POP STACK INTO R0
072166 000002          RTI
072170 005726          4$:  TST     (SP)+          ;;CLEAN PARTIAL FROM STACK
072172 105010          CLRB   (R0)             ;;SET A TERMINATOR
072174 104401          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
072176 000000          5$:  .WORD   0
072200 104401 001206  TYPE     ,$QUES          ;; '?' 'CR' & 'LF'
072204 000730          BR     1$              ;;TRY AGAIN
072206 000000          $HIOCT: .WORD 0      ;;HIGH ORDER BITS GO HERE
  
```

.SBTTL 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

```

```

072210          $SAVREG:
072210 010046      MOV     R0,-(SP)      ;;PUSH R0 ON STACK
072212 010146      MOV     R1,-(SP)      ;;PUSH R1 ON STACK
072214 010246      MOV     R2,-(SP)      ;;PUSH R2 ON STACK
072216 010346      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
072220 010446      MOV     R4,-(SP)      ;;PUSH R4 ON STACK
072222 010546      MOV     R5,-(SP)      ;;PUSH R5 ON STACK
072224 016646 000022 MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
072230 016646 000022 MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
072234 016646 000022 MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
072240 016646 000022 MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
072244 000002      RTI

```

*RESTORE R0-R5

```

:*CALL:
:*   RESREG

```

```

072246          $RESREG:
072246 012666 000022 MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
072252 012666 000022 MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
072256 012666 000022 MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
072262 012666 000022 MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
072266 012605      MOV     (SP)+,R5      ;;POP STACK INTO R5
072270 012604      MOV     (SP)+,R4      ;;POP STACK INTO R4
072272 012603      MOV     (SP)+,R3      ;;POP STACK INTO R3
072274 012602      MOV     (SP)+,R2      ;;POP STACK INTO R2
072276 012601      MOV     (SP)+,R1      ;;POP STACK INTO R1
072300 012600      MOV     (SP)+,R0      ;;POP STACK INTO R0
072302 000002      RTI

```


.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.

```

```

072304 010046
072306 016600 000002
072312 005740
072314 111000
072316 006300
072320 016000 072340
072324 000200

```

```

$TRAP:  MOV    RO,-(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

```

072326 011646
072330 016666 000004 000002
072336 000002

```

```

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

```

.SBTTL TRAP TABLE

```

:*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
:*BY THE 'TRAP' INSTRUCTION.

```

```

:      ROUTINE
:      -----
$TRPAD: .WORD  $TRAP2
        $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)

072340 072326      $GTSWR  ;;CALL=GTSWR      TRAP+6(104406)  GET SOFT-SWR SETTING

072354 071200
072356 071110      $CKSWR  ;;CALL=CKSWR     TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
072360 071452      $RDCHR  ;;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
072362 071542      $RDLIN  ;;CALL=RDLIN     TRAP+11(104411) TTY TYPEIN STRING ROUTINE
072364 072050      $RDOCT  ;;CALL=RDOCT     TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
072366 072210      $$AVREG ;;CALL=SAVREG     TRAP+13(104413) SAVE R0-R5 ROUTINE
072370 072246      $RESREG ;;CALL=RESREG     TRAP+14(104414) RESTORE R0-R5 ROUTINE

```

.SBTTL TELETYPE MESSAGES

2				
3	072372	200	105	116 ENTERA: .ASCIZ <CRLF>/ENTER DRIVE ADDRESS: /
4	072421	040	077	111 ADRERR: .ASCIZ / ?INVALID ADDRESS/<CRLF>
5	072444	200	120	117 PORTAIS: .ASCIZ <CRLF>/PORT 'A' ADDRESS IS: /
6	072473	200	120	117 PORTBIS: .ASCIZ <CRLF>/PORT 'B' ADDRESS IS: /
7	072522	200	116	117 NOCLOCK: .ASCIZ <CRLF>/NO SYSTEM 'L' OR 'P' CLOCK/<CRLF><LF>
8	072560	012	105	116 TESTNO: .ASCIZ <LF>/ENTER TEST #: /
9	072600	040	077	111 BADNO: .ASCIZ / ?INVALID TEST NUMBER/<CRLF>
10	072627	040	105	122 TSTERR: .ASCIZ / ERRORS/<CRLF>
11	072640	200	012	122 ADDRIS: .ASCIZ <CRLF><LF>@RH/RM ADDRESS (RMCS1) IS: @
12	072675	012	105	116 NTRH: .ASCIZ <LF>@ENTER RH/RM ADDRESS: @

```

1          .SBTTL TEST ERROR MESSAGES
2
3 072724    127    122    117 EM1:  .ASCIZ /WRONG DRIVE TYPE/
4 072745    104    122    111 EM2:  .ASCIZ /DRIVE NOT ON LINE/
5 072767    123    105    122 EM3:  .ASCIZ /SERIAL NUMBER READ THROUGH EACH PORT NOT THE SAME/
6 073051    104    122    111 EM4:  .ASCIZ /DRIVE NOT SEIZED BY PORT/
7 073102    127    122    117 EM5:  .ASCIZ /WRONG STATUS SEEN BY THE SEIZING PORT/
8 073150    122    105    107 EM6:  .ASCIZ /REGISTER CONTENTS WERE SEEN BY OPPOSITE PORT - DRIVE WAS SEIZED/
9 073250    122    105    107 EM7:  .ASCIZ /REGISTER CONTENTS WRONG AFTER RRELEASE OR TIMEOUT/
10 073331   122    105    107 EM10: .ASCIZ /REGISTER CONTENTS WRONG/
11 073361   103    117    116 EM11: .ASCIZ /CONTROL BUS PARITY ERROR READING INDICATED REGISTER/
12 073445   104    122    111 EM12: .ASCIZ /DRIVE NOT SEIZED BY DRIVE CLEAR COMMAND/
13 073515   122    105    101 EM13: .ASCIZ /READIN PRESET DOES NOT SET VOLUME VALID FOR THE PORT/
14 073602   126    117    114 EM14: .ASCIZ /VOLUME VALID SET ON THE WRONG PORT/
15 073645   101    124    124 EM15: .ASCIZ /ATTN BIT WRONG AFTER TIMEOUT - REQUEST NOT SET/
16 073724   101    124    124 EM16: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST SET/
17 073777   101    124    124 EM17: .ASCIZ /ATTN BIT WRONG AFTER RELEASE - REQUEST NOT SET/
18 074056   104    122    111 EM20: .ASCIZ /DRIVE NOT SEIZED WHEN ATTN BIT FOR PORT CLEARED/
19 074136   104    122    111 EM21: .ASCIZ /DRIVE SEIZED WHEN ZERO WRITTEN IN ATTN BIT/
20 074211   104    122    111 EM22: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER TIMEOUT - REQUEST NOT SET/
21 074276   124    111    115 EM23: .ASCIZ /TIMEOUT CLEARED THE DRIVE'S ERROR BIT/
22 074344   122    105    114 EM24: .ASCIZ /RELEASE COMMAND RELEASED DRIVE WITH ERRORS SET/
23 074423   124    111    115 EM25: .ASCIZ /TIMEOUT ONE-SHOT DID NOT RETRIGGER/
24 074466   104    122    111 EM26: .ASCIZ /DRIVE NOT IN NEUTRAL AFTER RELEASE - REQUEST NOT SET/
25 074553   122    105    107 EM27: .ASCIZ /REGISTER WRONG AFTER RELEASE WITH REQUEST SET/
26 074631   104    122    111 EM30: .ASCIZ /DRIVE SEIZED BY RELEASE COMMAND ISSUED WHEN DRIVE IN NEUTRAL/
27 074726   104    122    111 EM31: .ASCIZ /DRIVE IN NEUTRAL AFTER RELEASE - REQUEST SET/
28 075003   101    124    124 EM32: .ASCIZ /ATTN BIT WRONG AFTER RECALIBRATE COMMAND/
29 075054   104    122    111 EM33: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF DRIVE CLEAR GIVEN WHILE DRIVE SEIZED/
30 075156   104    122    111 EM34: .ASCIZ /DRIVE RETURNED TO NEUTRAL IF MASSBUS INIT GIVEN WHILE DRIVE SEIZED/
31 075261   124    111    115 EM35: .ASCIZ /TIMEOUT ONE SHOT FIRED WITHOUT REGISTER ACCESS/
32 075340   124    111    115 EM36: .ASCIZ /TIMEOUT HAS NOT OCCURRED WITHIN 2 SECONDS/
33 075412   104    122    111 EM37: .ASCIZ /DRIVE IS NON-EXISTENT ('NED' BIT SET)/
34 075460   101    124    124 EM40: .ASCIZ /ATTN BIT FOR PORT NOT RESET BY MASSBUS CLEAR/
35 075535   124    111    115 EM41: .ASCIZ /TIMEOUT CLEARED THE ATTENTION BIT/
36 075577   104    122    111 EM42: .ASCIZ /DRIVE NOT IN NEUTRAL OR SEIZED AFTER ATTN BIT WRITTEN/
37 075665   104    122    111 EM43: .ASCIZ /DRIVE IN NEUTRAL AFTER ATTENTION BIT WRITTEN/
38 075742   127    122    111 EM44: .ASCIZ /WRITE ATTENTION BIT DID NOT SET PORT REQUEST/
39 076017   120    117    122 EM45: .ASCIZ @PORT SELECT SWITCH ON DRIVE, NOT IN 'A/B'@
40 076071   103    101    116 EM46: .ASCIZ /CAN'T ACCESS DRIVE THROUGH EITHER PORT/
41 076140   101    124    124 EM47: .ASCIZ /ATTN BIT FOR SEIZING PORT NOT CLEARED BY MASSBUS INIT/
42 076226   101    124    124 EM50: .ASCIZ /ATTN BIT FOR OPPOSITE PORT CLEARED BY DRIVE CLEAR/
43 076310   101    124    124 EM51: .ASCIZ /ATTN BIT NOT CLEARED BY MASSBUS INIT, DRIVE IN NEUTRAL/
44 076377   124    110    105 EM52: .ASCIZ /THE ATTN BIT SET AFTER TIMEOUT WITH NO REQUEST & 'ERR' SET/
45 076472   103    101    116 EM53: .ASCIZ /CAN'T READ THE ATTN BIT FROM THE 'OPPOSITE' PORT/
46 076553   122    105    114 EM54: .ASCIZ /RELEASE COMMAND RECOGNIZED WHEN ISSUED BY NON-SEIZING PORT/
47 076646   124    111    115 EM55: .ASCIZ /TIMEOUT ONE-SHOT IS LESS THAN 500 MS/
48 076713   122    110    057 EM56: .ASCIZ @RH/RM DIDN'T RESPOND TO ADDRESSING@
49 076756   120    117    122 EM57: .ASCIZ /PORT REQUEST FLOP(S) WRONG STATE/
50 077017   101    124    124 EM60: .ASCIZ /ATTENTION NOT RESET BY WRITING RMAS/
51 077063   101    124    124 EM61: .ASCIZ /ATTENTION NOT RESET BY GO/
52 077115   101    124    124 EM62: .ASCIZ /ATTENTION RESET BY GO WHEN NOT SEIZED/
53 077163   104    122    111 EM63: .ASCIZ /DRIVE SEIZED BY UNIT READY CHANGE/
54 077225   101    124    124 EM64: .ASCIZ /ATTENTION NOT SET BY UNIT READY CHANGE/
55 077274   126    117    114 EM65: .ASCIZ /VOLUME VALID NOT RESET BY UNIT READY/

```

Line	Address	Port	PC	Device	Mode	Err PC	Port #	Reg ADR	Contents
1	077341	124	105	123	DH1:	.ASCIZ	/TEST #	ERR PC	PORT # REG ADR CONTENTS/
2	077412	124	105	123	DH3:	.ASCIZ	/TEST #	ERR PC	REG ADR PORT A PORT B/
3	077461	040	040	040	DH4:	.ASCII	/		SEIZE ERROR/<CR><LF>
4	077520	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # PORT # REG ADR GOOD BAD/
5	077604	124	105	123	DH5:	.ASCIZ	/TEST #	ERR PC	PORT # REG ADR GOOD BAD/
6	077660	040	040	040	DH7:	.ASCII	/		RELSNG ERROR/<CR><LF>
7	077717	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # PORT # REG ADR GOOD BAD/
8	100003	124	105	123	DH11:	.ASCIZ	/TEST #	ERR PC	PORT # REG ADR CONTENTS/
9	100054	040	040	040	DH13:	.ASCII	/		SEIZE ERROR/<CR><LF>
10	100113	124	105	123		.ASCIZ	/TEST #	ERR PC	PCRT # PORT # REG ADR CONTENTS/
11	100174	040	040	040	DH22:	.ASCII	/		RELSNG SEIZE/<CR><LF>
12	100233	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # PORT #
13	100272	040	040	040	DH23:	.ASCII	/		SEIZE/<CR><LF>
14	100321	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # REG ADR CONTENTS/
15	100372	040	040	040	DH26:	.ASCII	/		RELSNG/<CR><LF>
16	100422	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT #
17	100451	040	040	040	DH31:	.ASCII	/		RELSNG RQSTNG/<CR><LF>
18	100511	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # PORT #
19	100550	124	105	123	DH36:	.ASCIZ	/TEST #	ERR PC	PORT #
20	100577	124	105	123	DH42:	.ASCIZ	/TEST #	ERR PC/	
21	100616	040	040	040	DH44:	.ASCII	/		RELSNG ERROR/<CR><LF>
22	100655	124	105	123		.ASCIZ	/TEST #	ERR PC	PORT # PORT #
23	100714	040	040	040	DH46:	.ASCII	/		PORT A PORT B/<CR><LF>
24	100754	124	105	123		.ASCIZ	/TEST #	ERR PC	RMDS RMDS/
25	101011	124	105	123	DH55:	.ASCIZ	/TEST #	ERR PC	PORT # TIMEOUT VALUE (IN MS)/
26	101067	044	122	115	DH56:	.ASCIZ	/SRMADR/		
27	101076	124	105	123	DH57:	.ASCII	/TEST #	ERR PC	PORT A PORT B/
28	101150	040	040	040		.ASCIZ	/		EXPCTD RECEVD EXPECTD RECEVD/
29									
30									
31						.EVEN			
32	101230	001246	001116	001240	DT1:	.WORD	TSTNUM,\$ERRPC,\$PTNBR,\$BDADR,\$BDDAT,0		
33	101244	001246	001116	001122	DT3:	.WORD	TSTNUM,\$ERRPC,\$BDADR,\$GDDAT,\$BDDAT,0		
34	101260	001246	001116	001240	DT5:	.WORD	TSTNUM,\$ERRPC,\$PTNBR,\$BDADR,\$GDDAT,\$BDDAT,0		
35	101276	001246	001116	001244	DT6:	.WORD	TSTNUM,\$ERRPC,\$OPPRT,\$BDADR,\$BDDAT,0		
36	101312	001246	001116	001242	DT7:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$PTNBR,\$BDADR,\$GDDAT,\$BDDAT,0		
37	101332	001246	001116	001242	DT13:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$PTNBR,\$BDADR,\$BDDAT,0		
38	101350	001246	001116	001242	DT22:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$PTNBR,0		
39	101362	001246	001116	001242	DT23:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$BDADR,\$BDDAT,0		
40	101376	001246	001116	001242	DT31:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$OPPRT,0		
41	101410	001246	001116	001242	DT36:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,0		
42	101420	001246	001116	001240	DT37:	.WORD	TSTNUM,\$ERRPC,\$PTNBR,0		
43	101430	001246	001116	000000	DT42:	.WORD	TSTNUM,\$ERRPC,0		
44	101436	001246	001116	001170	DT46:	.WORD	TSTNUM,\$ERRPC,\$TMP2,\$TMP3,0		
45	101450	001246	001116	001244	DT54:	.WORD	TSTNUM,\$ERRPC,\$OPPRT,\$SEIZPT,0		
46	101462	001246	001116	001242	DT55:	.WORD	TSTNUM,\$ERRPC,\$SEIZPT,\$TIME,0		
47	101474	001304	000000		DT56:	.WORD	\$RMADR,0		
48	101500	001246	001116	001164	DT57:	.WORD	TSTNUM,\$ERRPC,\$TMP0,\$TMP1,\$TMP2,\$TMP3,0		
49									
50	101516	000	000	000	DF1:	.BYTE	0,0,0,0,0		
51	101523	000	000	000	DF5:	.BYTE	0,0,0,0,0,0		
52	101531	000	000	000	DF7:	.BYTE	0,0,0,0,0,0,0		
53	101540	000	000	000	DF31:	.BYTE	0,0,0,0		
54	101544	000	000	000	DF36:	.BYTE	0,0,0		
55	101547	000	000	000	DF42:	.BYTE	0,0		
56	101551	000	000	000	DF55:	.BYTE	0,0,0,1		
57	101555	000	000	000	DF56:	.BYTE	0		

58

.EVEN

```

1          .SBTTL  CONSTANTS, TABLES, ETC
2
3          :TABLE OF TEST STARTING ADDRESSES
4
5 101556 003374 TSTADR: .WORD TST1          ;STARTING ADDRESS OF TEST 1
8 101560 004746 .WORD TST2          ;STARTING ADDRESS OF TEST 2
 101562 006314 .WORD TST3          ;STARTING ADDRESS OF TEST 3
 101564 007662 .WORD TST4          ;STARTING ADDRESS OF TEST 4
 101566 011012 .WORD TST5          ;STARTING ADDRESS OF TEST 5
 101570 012142 .WORD TST6          ;STARTING ADDRESS OF TEST 6
 101572 012614 .WORD TST7          ;STARTING ADDRESS OF TEST 7
 101574 013266 .WORD TST10         ;STARTING ADDRESS OF TEST 10
 101576 014532 .WORD TST11         ;STARTING ADDRESS OF TEST 11
 101600 015776 .WORD TST12         ;STARTING ADDRESS OF TEST 12
 101602 017116 .WORD TST13         ;STARTING ADDRESS OF TEST 13
 101604 020236 .WORD TST14         ;STARTING ADDRESS OF TEST 14
 101606 021636 .WORD TST15         ;STARTING ADDRESS OF TEST 15
 101610 023236 .WORD TST16         ;STARTING ADDRESS OF TEST 16
 101612 024162 .WORD TST17         ;STARTING ADDRESS OF TEST 17
 101614 025106 .WORD TST20         ;STARTING ADDRESS OF TEST 20
 101616 026154 .WORD TST21         ;STARTING ADDRESS OF TEST 21
 101620 027222 .WORD TST22         ;STARTING ADDRESS OF TEST 22
 101622 031276 .WORD TST23         ;STARTING ADDRESS OF TEST 23
 101624 032022 .WORD TST24         ;STARTING ADDRESS OF TEST 24
11 101626 033216 .WORD TST25         ;STARTING ADDRESS OF TEST 25
 101630 034412 .WORD TST26         ;STARTING ADDRESS OF TEST 26
 101632 036106 .WORD TST27         ;STARTING ADDRESS OF TEST 27
 101634 037602 .WORD TST30         ;STARTING ADDRESS OF TEST 30
 101636 041276 .WORD TST31         ;STARTING ADDRESS OF TEST 31
 101640 042772 .WORD TST32         ;STARTING ADDRESS OF TEST 32
 101642 044250 .WORD TST33         ;STARTING ADDRESS OF TEST 33
 101644 045374 .WORD TST34         ;STARTING ADDRESS OF TEST 34
 101646 046272 .WORD TST35         ;STARTING ADDRESS OF TEST 35
14 101650 047170 .WORD TST36         ;STARTING ADDRESS OF TEST 36
 101652 050166 .WORD TST37         ;STARTING ADDRESS OF TEST 37
 101654 051164 .WORD TST40         ;STARTING ADDRESS OF TEST 40
 101656 053150 .WORD TST41         ;STARTING ADDRESS OF TEST 41
 101660 055134 .WORD TST42         ;STARTING ADDRESS OF TEST 42
 101662 056410 .WORD TST43         ;STARTING ADDRESS OF TEST 43
 101664 057664 .WORD TST44         ;STARTING ADDRESS OF TEST 44
 101666 060514 .WORD TST45         ;STARTING ADDRESS OF TEST 45
 101670 061344 .WORD TST46         ;STARTING ADDRESS OF TEST 46
    
```

```

15
16          :ATTENTION BIT TABLE
17
18 101672 001  ATABIT: .BYTE 1          ;ATTENTION BIT FOR DRIVE 0
19 101673 002 .BYTE 2          ;ATTENTION BIT FOR DRIVE 1
20 101674 004 .BYTE 4          ;ATTENTION BIT FOR DRIVE 2
21 101675 010 .BYTE 10         ;ATTENTION BIT FOR DRIVE 3
22 101676 020 .BYTE 20         ;ATTENTION BIT FOR DRIVE 4
23 101677 040 .BYTE 40         ;ATTENTION BIT FOR DRIVE 5
24 101700 100 .BYTE 100        ;ATTENTION BIT FOR DRIVE 6
25 101701 200 .BYTE 200        ;ATTENTION BIT FOR DRIVE 7
26
29 101702 000052 MAXTN: .WORD 52      ;MAXIMUM TEST NUMBER
30
31          .END 200
    
```

ADDRIS 072640	CR = 000015	DT42 101430	EM56 076713	OPE = 020000
ADRERR 072421	CRLF = 000200	DT46 101436	EM57 076756	OPI = 020000
AGE = 001000	DCK = 100000	DT5 101260	EM6 073150	OPPRT 001244
ASR1 001236	DDISP = 177570	DT54 101450	EM60 077017	OR = 000200
ATA = 100000	DF1 101516	DT55 101462	EM61 077063	PAR = 000010
ATABIT 101672	DF31 101540	DT56 101474	EM62 077115	PAT = 000020
ATO = 000001	DF36 101544	DT57 101500	FM63 077163	PFECH 067472
AT1 = 000002	DF42 101547	DT6 101276	EM64 077225	PFECH1 067502
AT2 = 000004	DF5 101523	DT7 101312	EM65 077274	PFECH2 067564
AT3 = 000010	DF55 101551	DVA = 004000	EM7 073250	PFECH3 067616
AT4 = 000020	DF56 101555	DVC = 000200	ENTERA 072372	PFECH4 067626
AT5 = 000040	DF7 101531	ECH = 000100	ERR = 040000	PFTSTN 067632
AT6 = 000100	DH1 077341	ECI = 004000	ERROR = 104000	PGE = 002000
AT7 = 000200	DH11 100003	EMTVEC = 000030	ERRVEC = 000004	PGM = 001000
A16 = 000400	DH13 100054	EM1 072724	EXEC = 003074	PIP = 020000
A17 = 001000	DH22 100174	EM10 073331	FER = 000020	PIRQ = 177772
BADNO 072600	DH23 100272	EM11 073361	FMT16 = 010000	PIRQVE = 000240
BADTMO 002160	DH26 100372	EM12 073445	F0 = 000002	PORTA 001224
BAI = 000010	D43 077412	EM13 073515	F1 = 000004	PORTAI 072444
BIT0 = 000001	DH31 100451	EM14 073602	F2 = 000010	PORTB 001226
BIT00 = 000001	DH36 100550	EM15 073645	F3 = 000020	PORTBI 072473
BIT01 = 000002	DH4 077461	EM16 073724	F4 = 000040	PORTC 001230
BIT02 = 000004	DH42 100577	EM17 073777	GO = 000001	PR0 = 000000
BIT03 = 000010	DH44 100616	EM2 072745	GTSWR = 104406	PR1 = 000040
BIT04 = 000020	DH46 100714	EM20 074056	HCE = 000200	PR2 = 000100
BIT05 = 000040	DH5 077604	EM21 074136	HCI = 002000	PR3 = 000140
BIT06 = 000100	DH55 101011	EM22 074211	HCRC = 000400	PR4 = 000200
BIT07 = 000200	DH56 101067	EM23 074276	HT = 000011	PR5 = 000240
BIT08 = 000400	DH57 101076	EM24 074344	IAE = 002000	PR6 = 000300
BIT09 = 001000	DH7 077660	EM25 074423	IBSAVE 067272	PR7 = 000340
BIT1 = 000002	DISPLA 001142	EM26 074466	IE = 000100	PS = 177776
BIT10 = 002000	DISPRE 000174	EM27 074553	ILF = 000001	PSEL = 002000
BIT11 = 004000	DLT = 100000	EM3 072767	ILR = 000002	PSW = 177776
BIT12 = 010000	DMD = 000001	EM30 074631	IOTVEC = 000020	PTNBR 001240
BIT13 = 020000	DPE = 000010	EM31 074726	IR = 000100	PWRVEC = 000024
BIT14 = 040000	DPR = 000400	EM32 075003	IVC = 010000	RDCHR = 104410
BIT15 = 100000	DRQ = 004000	EM33 075054	KYBCTL 001300	RDLIN = 104411
BIT2 = 000004	DRY = 000200	EM34 075156	LBC = 002000	RDOCT = 104412
BIT3 = 000010	DSWR = 177570	EM35 075261	LBT = 002000	RDY = 000200
BIT4 = 000020	DTE = 010000	EM36 075340	LF = 000012	RELERR 001254
BIT5 = 000040	DT00 = 000001	EM37 075412	LSC = 004000	RELOK = 000001
BIT6 = 000100	DT01 = 000002	EM4 073051	MAXTN 101702	RESREG = 104414
BIT7 = 000200	DT02 = 000004	EM40 075460	MCPE = 020000	RESVEC = 000010
BIT8 = 000400	DT03 = 000010	EM41 075535	MDPE = 000400	RMAS = 000016
BIT9 = 001000	DT04 = 000020	EM42 075577	MOH = 020000	RMBA = 000004
BPTVEC = 000014	DT05 = 000040	EM43 075665	MOL = 010000	RMCS1 = 000000
CHANGE 003246	DT06 = 000100	EM44 075742	MUR = 001000	RMCS2 = 000010
CHGADR 001302	DT07 = 000200	EM45 076017	MXF = 001000	RMDA = 000006
CKCLK 066130	DT08 = 000400	EM46 076071	NBA = 100000	RMDB = 000022
CKCLK1 066200	DT1 101230	EM47 076140	NED = 010000	RMDC = 000034
CKCLK2 066242	DT13 101332	EM5 073102	NEM = 004000	RMDS = 000012
CKCLK3 066252	DT22 101350	EM50 076226	NOATA = 000001	RMDT = 000026
CKERR 001250	DT23 101362	EM51 076310	NOCLOC 072522	RMEC1 = 000044
CKSWR = 104407	DT3 101244	EM52 076377	NOSEIZ 001252	RMEC2 = 000046
CLOCK 066262	DT31 101376	EM53 076472	NTRH 072675	RMER1 = 000014
CLR = 000040	DT36 101410	EM54 076553	OFD = 000200	RMER2 = 000042
CPSAVE 067270	DT37 101420	EM55 076646	OM = 000001	RMLA = 000020

RMMR1 = 000024	TBITVE= 000014	TSTADR 101556	U3 = 000004	\$LPERR 001110
RMMR2 = 000040	TESTNO 072560	TSTERR 072627	VV = 000100	\$LPVEC 001216
RMOF = 000032	TEST1 003432	TSTNUM 001246	VVSET = 000001	\$MNEW 072036
RMR = 000004	TEST10 013324	TST1 003374	WATCH = 001260	\$MSWR 072025
RMSN = 000030	TEST11 014570	TST1AA 003356	WCE = 040000	\$MXCNT 066716
RMWC = 000002	TEST12 016034	TST10 013266	WCF = 000040	\$NULL 001154
RQA = 100000	TEST13 017154	TST11 014532	WLE = 004000	\$NWTST= 000000
RQB = 040000	TEST14 020274	TST12 015776	WRL = 004000	\$OCNT 070350
RQSTA 001232	TEST15 021674	TST13 017116	\$AUTOB 001134	\$OMODE 070352
RQSTB 001234	TEST16 023274	TST14 020236	\$BDADR 001122	\$OVER 066702
R6 = %000006	TEST17 024220	TST15 021636	\$BDDAT 001126	\$PASS 001100
R7 = %000007	TEST2 005004	TST16 023236	\$BELL 001202	\$PASSES 001206
SAVREG= 104413	TEST20 025144	TST17 024162	\$CHARC 070122	\$RDCHR 071452
SC = 100000	TEST21 026212	TST2 004746	\$CKSWR 071110	\$RDLIN 071542
SCOPE = 000004	TEST22 027260	TST20 025106	\$CMTAG 001100	\$RDOCT 072050
SCO = 000100	TEST23 031334	TST21 026154	\$CM1 = 000001	\$RDSZ = 000010
SC1 = 000200	TEST24 032060	TST22 027222	\$CM2 = 000002	\$REGAD 001160
SC2 = 000400	TEST25 033254	TST23 031276	\$CM3 = 000001	\$REGO 001162
SC3 = 001000	TEST26 034450	TST24 032022	\$CM4 = 000005	\$RESRE 072246
SC4 = 002000	TEST27 036144	TST25 033216	\$CNTLC 072006	\$RMADR 001304
SEIZPT 001242	TEST3 006352	TST26 034412	\$CNTLG 072020	\$RMVEC 001306
SKI = 100000	TEST30 037640	TST27 036106	\$CNTLU 072013	\$RTNAD 066122
STACK = 001100	TEST31 041334	TST3 006314	\$CRLF 001207	\$SAVRE 072210
START 002240	TEST32 043030	TST30 037602	\$DBLK 070570	\$SCOPE 066364
START1 002246	TEST33 044306	TST31 041276	\$DOAGN 066120	\$SETUP= 000127
START2 002254	TEST34 045432	TST32 042772	\$DTBL 070560	\$STUP = 177777
STKLMT= 177774	TEST35 046330	TST33 044250	\$ENDAD 066110	\$SVLAD 066672
STOP 066720	TEST36 047226	TST34 045374	\$ENDCT 065746	\$SVPC = 000210
SWR 001140	TEST37 050224	TST35 046272	\$ENULL 066124	\$SWR = 166000
SWREG 000176	TEST4 007720	TST36 047170	\$EOP 065702	\$SWRMK= 000000
SW0 = 000001	TEST40 051222	TST37 050166	\$EOPCT 065740	\$TIMES 001176
SW00 = 000001	TEST41 053206	TST4 007662	\$ERFLG 001103	\$TKB 001146
SW01 = 000002	TEST42 055172	TST40 051164	\$ERMAX 001115	\$TKCNT 070600
SW02 = 000004	TEST43 056446	TST41 053150	\$ERROR 066746	\$TKINT 070610
SW03 = 000010	TEST44 057722	TST42 055134	\$ERRPC 001116	\$TKQEN= 070607
SW04 = 000020	TEST45 060552	TST43 056410	\$ERRTB 001310	\$TKQIN 070602
SW05 = 000040	TEST46 061402	TST44 057664	\$ERRTY 067274	\$TKQOU 070604
SW06 = 000100	TEST47 062266	TST45 060514	\$ERTTL 001112	\$TKQSR 070606
SW07 = 000200	TEST5 011050	TST46 061344	\$ESCAP 001200	\$TKS 001144
SW08 = 000400	TEST50 063152	TST47 062230	\$FILLC 001156	\$TKSRV 070660
SW09 = 001000	TEST51 064444	TST5 011012	\$FILLS 001155	\$TMP0 001164
SW1 = 000002	TEST6 012200	TST50 063114	\$GDADR 001120	\$TMP1 001166
SW10 = 002000	TEST7 012652	TST50B 064164	\$GDDAT 001124	\$TMP2 001170
SW11 = 004000	TIME 001256	TST51 064406	\$GET42 066100	\$TMP3 001172
SW12 = 010000	TIMEA 001262	TST51B 065456	\$GT5WR 071200	\$TMP4 001174
SW13 = 020000	TIMEAM 001266	TST52 065700	\$GT42P 066074	\$TN = 000053
SW14 = 040000	TIMEAP 001264	TST6 012142	\$HD = 000000	\$TPB 001152
SW15 = 100000	TIMEB 001270	TST7 012614	\$HIOCT 072206	\$TPFLG 001157
SW2 = 000004	TIMEBM 001274	TYPDS = 104405	\$ICNT 001104	\$TPS 001150
SW3 = 000010	TIMEBP 001272	TYPE = 104401	\$INTAG 001135	\$TRAP 072304
SW4 = 000020	TIMES 001276	TYPOC = 104402	\$ITEMB 001114	\$TRAP2 072326
SW5 = 000040	TKVEC = 000060	TYPON = 104404	\$LF 001210	\$TRP = 000015
SW6 = 000100	TOLER 066324	TYPOS = 104403	\$LKCSB 001214	\$TRPAD 072340
SW7 = 000200	TPVEC = 000064	UNS = 040000	\$LKCSR 001212	\$TSTM 001102
SW8 = 000400	TRAPVE= 000034	UPE = 020000	\$LKS 001220	\$TTYIN 071776
SW9 = 001000	TRE = 040000	U0 = 000001	\$LLVEC 001222	\$TYPDS 070354
TAP = 040000	TRTVEC= 000014	U1 = 000002	\$LPADR 001106	\$TYPE 067634

\$TYPEC 070004
\$TYPEX 070124

\$TYPOC 070152
\$TYPON 070166

\$TYPOS 070126
\$XOFF = 000023

\$XON = 000021
\$XTSTR 066406

\$GET4= 000000
\$FILL 070351

. ABS. 101704 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 61952 WORDS (242 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
CZMRB.BIN,CZMRB/C=CZMRB.DOC,CZMRB,SYSMAC/M

SERTY 14-1 15-1#

C 5

SEO 0199

\$ERTTL	6-0#	11-1	11-1	11-1*	14-1	14-1	14-1*							
\$ESCAP	6-0#	9-24*												
\$FILLC	6-0#	16-1	16-1	16-1										
\$FILLS	6-0#	16-1	16-1											
\$GDADR	6-0#													
\$GDDAT	6-0#	10-33	10-33	10-33*	10-33*	10-33*	10-33*	10-40	10-40	10-40*	10-40*	10-48	10-48	10-48
	10-48	10-48*	10-48*	10-48*	10-48*	10-48*	10-48*	10-48*	10-48*	10-53*	10-56	10-151	10-151	10-151
	10-151	10-151	10-151*	10-151*	10-151*	10-173	10-173	10-173	10-173	10-173	10-173*	10-173*	10-173*	10-223
	10-223	10-223	10-223	10-223	10-223*	10-223*	10-223*	10-223*	10-223*	10-223*	10-223*	10-241	10-241	10-241
	10-241	10-241	10-241*	10-241*	10-241*	10-241*	10-241*	10-241*	10-241*	10-268	10-268	10-268*	10-281	10-281
	10-281*	10-327	10-327	10-327	10-327	10-327	10-327	10-327	10-327	10-327*	10-327*	10-327*	10-327*	10-350
	10-350	10-350	10-350	10-350	10-350	10-350	10-350	10-350	10-350*	10-350*	10-350*	10-407	10-407	10-407
	10-407	10-407	10-407*	10-407*	10-407*	10-407*	10-407*	10-407*	10-431	10-431	10-431	10-431	10-431	10-431*
	10-431*	10-431*	10-431*	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483*
	10-483*	10-483*	10-483*	10-483*	10-483*	10-483*	10-483*	10-483*	10-483*	10-506	10-506	10-506	10-506	10-506
	10-506	10-506	10-506	10-506	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*
	10-506*	10-531	10-531	10-531	10-531	10-531	10-531*	10-531*	10-531*	10-531*	10-547	10-547	10-547	10-547
	10-547*	10-547*	10-547*	10-583	10-583	10-583	10-583	10-583	10-583	10-583	10-583*	10-583*	10-583*	10-583*
	10-603	10-603	10-603	10-603	10-603	10-603	10-603*	10-603*	10-603*	10-603*	10-603*	10-675	10-675	10-675
	10-675	10-675	10-675	10-675	10-675	10-675	10-675*	10-675*	10-675*	10-675*	10-675*	10-675*	10-675*	10-705
	10-705	10-705*	10-756	10-756	10-756	10-756	10-756	10-756*	10-756*	10-756*	10-756*	10-756*	10-756*	10-779
	10-779	10-779	10-779	10-779	10-779*	10-779*	10-779*	10-779*	10-779*	10-779*	10-779*	10-837	10-837	10-837
	10-837	10-837	10-837	10-837	10-837	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*
	10-837*	10-837*	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859*	10-859*
	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*
	10-914	10-914	10-914	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*
	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937*	10-937*	10-937*	10-937*
	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*
	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989	10-989	10-989	10-989	10-989*
	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989	10-989	10-989	10-989	10-989*
	10-:53*	10-:53*	10-:53*	10-:53*	10-:03	10-:03	10-:03	10-:03*	10-:03*	10-:03*	10-:03*	10-:20	10-:20	10-:20*
	10-:69	10-:69	10-:69	10-:69	10-:69*	10-:69*	10-:69*	10-:69*	10-:69*	10-:69*	10-:69*	10-:89	10-:89	10-:89*
	10-:89*	10-:89*	10-:89*	10-:89*	10-:84	10-:84	10-:84	10-:84*	10-:84*	10-:84*	10-:84*	10-:84	10-:84	10-:84*
	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*
	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09*	10-=09*	10-=09*
	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*
	10-=81	10-=81	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81	10-=81	10-=81
	10-=99	10-=99	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99	10-=99	10-=99
	10->73	10->73*	10-?41	10-?41	10-?41	10-?41*	10-?41*	10-?41*	10-?41*	10-?41*	10-?41*	10->53	10->53	10->53*
	10-a30	10-a30*	10-a30*	10-a31	10-a31	10-a31*	10-a31*	10-a31*	10-a31*	10-a31*	10-a31*	10->63	10->63	10->63*
	10-a61*	10-a61*	25-33	25-34	25-36							10-a60	10-a60	10-a60
\$GET42	11-1#													
\$GT42P	11-1	11-1#												
\$GTSWR	19-1#	22-1	22-1											
\$HD	4-598	4-598	4-598											
\$HIOCT	20-1#	20-1*												
\$ICNT	6-0#	9-93*	13-1	13-1	13-1	13-1*	13-1*							
\$INTAG	6-0#	19-1	19-1	19-1	19-1	19-1*								
\$ITEMB	6-0#	14-1	14-1	14-1	14-1	14-1*	14-1*	15-1						
\$LF	6-0#	14-1	14-1	16-1	16-1	19-1	19-1	19-1	20-1	20-1				
\$LKCSB	7-0#	12-12*												
\$LKCSR	7-0#	12-8	12-13*											
\$LKS	7-0#	12-17	12-21*											
\$LLVEC	7-0#	12-18												
\$LPADR	6-0#	9-24*	9-78*	10-20*	10-82*	10-171*	10-189*	10-239*	10-256*	10-279*	10-302*	10-348*	10-372*	10-429*
	10-452*	10-504*	10-520*	10-545*	10-565*	10-601*	10-620*	10-689*	10-726*	10-777*	10-803*	10-857*	10-880*	10-935*
	10-953*	10-:08*	10-:68*	10-:18*	10-:39*	10-:87*	10-<41*	10-=07*	10-=29*	10-=97*	10->17*	10->71*	10->96*	10-?61*

10-290* 10-258* 13-1 13-1 13-1 13-1*

E 5

SEQ 0201

10-583* 10-583* 10-583* 10-583* 10-603 10-603 10-603¹ 5 10-603* 10-603* 10-603* 10-603* 10-675 10-675 10-675
SEQ 0205

10--81* 10--81* 10--81* 10--81* 10--99 10--99 10--99^K 10--99⁵* 10--99* 10--99* 10--99* 10--99* 10--99* 10--99*

SE0 0207

\$TN	4-595#	4-598	10-20	10-20	10-20	10-20#	10-82	10-82	10-82	10-82#	10-171	10-171	10-171	10-171#
	10-189	10-189	10-189	10-189#	10-239	10-239	10-239	10-239#	10-256	10-256	10-256	10-256#	10-279	10-279
	10-279	10-279#	10-302	10-302	10-302	10-302#	10-348	10-348	10-348	10-348#	10-372	10-372	10-372	10-372#
	10-429	10-429	10-429	10-429#	10-452	10-452	10-452	10-452#	10-504	10-504	10-504	10-504#	10-520	10-520
	10-520	10-520#	10-545	10-545	10-545	10-545#	10-565	10-565	10-565	10-565#	10-601	10-601	10-601	10-601#
	10-620	10-620	10-620	10-620#	10-689	10-689	10-689	10-689#	10-726	10-726	10-726	10-726#	10-777	10-777
	10-777	10-777#	10-803	10-803	10-803	10-803#	10-857	10-857	10-857	10-857#	10-880	10-880	10-880	10-880#
	10-935	10-935	10-935	10-935#	10-953	10-953	10-953	10-953#	10-:08	10-:08	10-:08	10-:08#	10-:68	10-:68
	10-:68	10-:68#	10-:18	10-:18	10-:18	10-:18#	10-:39	10-:39	10-:39	10-:39#	10-:87	10-:87	10-:87	10-:87#
	10-<41	10-<41	10-<41	10-<41#	10-=07	10-=07	10-=07	10-=07#	10-=29	10-=29	10-=29	10-=29#	10-=97	10-=97
	10-=97	10-=97#	10->17	10->17	10->17	10->17#	10->71	10->71	10->71	10->71#	10->96	10->96	10->96	10->96#
	10-?61	10-?61	10-?61	10-?61#	10-?90	10-?90	10-?90	10-?90#	10-a30	10-a31	10-a31	10-a58	10-a58	10-a58
	10-a58#	10-a60	10-a61	10-a61	10-a65	10-a65	10-a65#	26-27						
\$TPB	6-0#	16-1	16-1	16-1*										
\$TPFLG	6-0#	16-1	16-1	16-1										
\$TPS	6-0#	16-1	16-1	16-1										
\$TRAP	9-24	22-1#												
\$TRAP2	22-1	22-1#												
\$TRP	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1
	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1
	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1	22-1
	22-1	22-1	22-1	22-1	22-1	22-1	22-1#	22-1#	22-1#	22-1#	22-1#	22-1#	22-1#	22-1#
\$TRPAD	22-1#	22-1#	22-1#	22-1#	22-1#									
\$STNM	6-0#	10-20*	10-82*	10-171*	10-189*	10-239*	10-256*	10-279*	10-302*	10-348*	10-372*	10-429*	10-452*	10-504*
	10-520*	10-545*	10-565*	10-601*	10-620*	10-689*	10-726*	10-777*	10-803*	10-857*	10-880*	10-935*	10-953*	10-:08*
	10-:68*	10-:18*	10-:39*	10-:87*	10-<41*	10-=07*	10-=29*	10-=97*	10->17*	10->71*	10->96*	10-?61*	10-?90*	10-a58*
	11-1*	13-1	13-1	13-1	13-1	13-1*	14-1	14-1	14-1	14-1	15-1			
\$TTYIN	19-1	19-1	19-1	19-1	19-1	19-1#								
\$TYPBN	22-1													
\$TYPDS	18-1#	22-1	22-1											
\$TYPE	16-1#	22-1	22-1											
\$TYPEC	16-1	16-1	16-1	16-1#	19-1									
\$TYPEX	16-1	16-1	16-1#											
\$TYPOC	17-1#	22-1	22-1											
\$TYPON	17-1	17-1#	22-1											
\$TYPOS	17-1#	22-1												
\$XOFF	16-1	16-1												
\$XON	16-1	16-1	19-1											
\$XTSTR	13-1#													
A16	4-609#													
A17	4-610#													
ADDRIS	9-102	23-11#												
ADRERR	9-38	23-4#												
AOE	4-679#													
ASR1	7-0#	9-55*	10-675	10-705	10-:53	10-:53	10-:53	10-:03	10-:03	10-:20	10-:20	10-a30	10-a60	
ATO	4-696#													
AT1	4-697#													
AT2	4-698#													
AT3	4-699#													
AT4	4-700#													
AT5	4-701#													
AT6	4-702#													
AT7	4-703#													
ATA	4-666#	10-151	10-151	10-151	10-151	10-173	10-173	10-173	10-173	10-223	10-223	10-223	10-223	10-223
	10-223	10-223	10-223	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-268	10-268	10-268

10-268 10-281 10-281 10-281 10-281 10-327 10-327^M 5 10-327 10-327 10-327 10-350 10-350 10-350 10-350
SEQ 0209

	10-350	10-407	10-407	10-407	10-407	10-431	10-431	10-431	10-431	10-483	10-483	10-483	10-483	10-506
	10-506	10-506	10-506	10-531	10-531	10-531	10-531	10-547	10-547	10-547	10-547	10-583	10-583	10-583
	10-583	10-603	10-603	10-603	10-603	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675
	10-675	10-705	10-705	10-705	10-705	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-779
	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-837	10-837	10-837	10-837	10-837	10-837	10-837
	10-837	10-837	10-837	10-837	10-837	10-837	10-837	10-837	10-837	10-859	10-859	10-859	10-859	10-859
	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859	10-859
	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914	10-914
	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937	10-937
	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989	10-989
	10-:53	10-:53	10-:53	10-:03	10-:03	10-:03	10-:03	10-:20	10-:20	10-:20	10-:20	10-:69	10-:69	10-:69
	10-:69	10-:69	10-:69	10-:69	10-:89	10-:89	10-:89	10-:89	10-:89	10-:89	10-:89	10-<84	10-<84	10-<84
	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84	10-<84
	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09	10-=09
	10-=81	10-=81	10-=81	10-=81	10-=81	10-=81	10-=81	10-=99	10-=99	10-=99	10-=99	10-=99	10-=99	10-=99
	10-=99	10->53	10->53	10->73	10->73	10-?41	10-?41	10-?41	10-?41	10-?63	10-?63	10-?63	10-?63	10-?63
	10-a30	10-a30	10-a60	10-a60	10-a60									
ATABIT	9-55	26-18#												
BADNO	9-87	23-9#												
BADTMO	9-3#	9-26												
BAI	4-621#													
BIT0	4-601#													
BIT00	4-601	4-601#	4-656	13-1	13-1	14-1	14-1							
BIT01	4-601	4-601#												
BIT02	4-601	4-601#												
BIT03	4-601	4-601#												
BIT04	4-601	4-601#												
BIT05	4-601	4-601#												
BIT06	4-601	4-601#												
BIT07	4-601	4-601#												
BIT08	4-601	4-601#												
BIT09	4-601	4-601#	4-690	13-1	14-1									
BIT1	4-601#													
BIT10	4-601#	14-1												
BIT11	4-601#	13-1												
BIT12	4-601#													
BIT13	4-601#	14-1												
BIT14	4-601#	4-691	13-1											
BIT15	4-601#	4-692												
BIT2	4-601#													
BIT3	4-601#													
BIT4	4-601#													
BIT5	4-601#	4-652												
BIT6	4-601#													
BIT7	4-601#													
BIT8	4-601#													
BIT9	4-601#													
BPTVEC	4-601#													
CHANGE	9-32	9-99#												
CHGADR	7-0#	9-14*	9-17*	9-99	9-101*									
CKCLK	9-59	9-72	12-6#											
CKCLK1	12-6	12-15#												
CKCLK2	12-16	12-23#												
CKCLK3	12-14	12-22	12-25#											
CKERR	7-0#	10-33	10-33	10-33*	10-33*	10-33*	10-33*	10-40*	10-40*	10-40*	10-40*	10-48*	10-48*	10-48*
	10-48*	10-48*	10-48*	10-48*	10-48*	10-223*	10-223*	10-223*	10-223*	10-223*	10-223*	10-241*	10-241*	10-241*

10-241• 10-241• 10-241• 10-327 10-327• 10-327• 10-350^B 6 10-350• 10-350• 10-407 10-407• 10-407• 10-407• 10-407• 10-407•
SEQ 0211

	10-407*	10-407*	10-431	10-431*	10-431*	10-431*	10-431*	10-431*	10-431*	10-431*	10-483*	10-483*	10-483*	10-483*	10-483*
	10-483*	10-483*	10-483*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-506*	10-583*	10-583*	10-603*
	10-603*	10-675*	10-675*	10-675*	10-675*	10-675*	10-756*	10-756*	10-756*	10-756*	10-779*	10-779*	10-779*	10-779*	10-837
	10-837	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-837*	10-859	10-859	10-859*	10-859*	10-859*
	10-859*	10-859*	10-859*	10-859*	10-859*	10-859*	10-914	10-914	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*	10-914*
	10-914*	10-937	10-937	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-937*	10-989	10-989	10-989*
	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*	10-989*
	10-:69*	10-:69*	10-:69*	10-:89*	10-:89*	10-:89*	10-:89*	10-:89*	10-:89*	10-:89*	10-:53	10-:53	10-:53*	10-:53*	10-:53*
	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*	10-<84*
	10-=09*	10-=09*	10-=09*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=81*	10-=09*	10-=09*	10-=09*	10-=09*	10-=09*
	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-=99*	10-?4	10-?41*	10-?41*	10-?41*	10-?63	10-?63*	10-?63*	10-?63*	10-?63*
	10-@31*	10-@61*	10-@61*	10-@61*	10-@61*	10-@61*									
CKSWR	13-1	14-1	14-1	22-1#											
CLOCK	12-10	12-19	12-30#												
CLP	4-623#	4-652#	10-22	10-33	10-33	10-223	10-241	10-407	10-431	10-483	10-506	10-989			
CPSAVE	13-1	13-1	13-1	13-1*	13-1*	14-1	14-1	14-1	14-1	14-1	14-1#	14-1*	14-1*	14-1*	15-1
CR	4-601#	16-1	16-1	25-3	25-6	25-9	25-11	25-13	25-15	25-17	25-21	25-23			
CRLF	4-601#	9-6	9-29	9-29	16-1	16-1	23-3	23-4	23-5	23-6	23-7	23-7	23-9	23-10	
	23-11														
DCK	4-685#														
DDISP	4-601#	6-0	9-24												
DF1	8-7	8-14	8-21	8-63	8-133	8-140	8-226	8-233	8-261	8-275	8-303	8-346	25-50#		
DF31	8-126	8-156	8-177	8-254	8-268	8-310	25-53#								
DF36	8-70	8-112	8-119	8-148	8-170	8-191	8-198	8-205	8-212	8-219	8-360	25-54#			
DF42	8-240	8-247	25-55#												
DF5	8-35	8-42	8-56	8-77	8-84	8-184	8-282	8-289	8-296	8-332	8-339	8-353	8-374	25-51#	
DF55	8-317	25-56#													
DF56	8-324	25-57#													
DF7	8-28	8-49	8-91	8-98	8-105	8-163	8-367	25-52#							
DH1	8-5	8-12	8-224	8-259	25-1#										
DH11	8-61	25-8#													
DH13	8-40	8-75	8-82	8-280	8-294	8-351	25-9#								
DH22	8-124	8-154	8-308	25-11#											
DH23	8-131	8-138	8-231	8-273	8-301	8-344	25-13#								
DH26	25-15#														
DH3	8-19	25-2#													
DH31	8-175	25-17#													
DH36	8-68	8-110	8-117	8-146	8-168	8-189	8-196	8-203	8-210	8-217	8-358	25-19#			
DH4	8-26	25-3#													
DH42	8-238	8-245	25-20#												
DH44	8-252	25-21#													
DH46	8-266	25-23#													
DH5	8-33	8-54	8-182	8-287	8-337	8-372	25-5#								
DH55	8-315	25-25#													
DH56	8-322	25-26#													
DH57	8-330	25-27#													
DH7	8-47	8-89	8-96	8-103	8-161	8-365	25-6#								
DISPLA	6-0#	9-24*	9-24*	13-1*	14-1*										
DISPRE	5-1#	9-24													
DLT	4-633#														
DMD	4-689#	10-<84	10-<84	10-=09	10-=09										
DPE	4-734#														
DPR	4-659#	10-48	10-48	10-151	10-151	10-173	10-173	10-223	10-241	10-268	10-281	10-327	10-327	10-327	10-327
	10-350	10-350	10-350	10-407	10-431	10-483	10-483	10-483	10-483	10-506	10-506	10-506	10-506	10-506	10-531
	10-531	10-547	10-547	10-583	10-583	10-603	10-603	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-705
	10-756	10-756	10-779	10-779	10-837	10-837	10-837	10-859	10-859	10-914	10-914	10-937	10-937	10-:03	10-:20

10-;69 10-;89 10-<84 10-<84 10-<84 10--09 10-=09^D 6 10-=09 10->53 10->73 10-?41 10-?63 10-a30 10-a30
SEQ 0213

EM26 8-153 24-24#

SEQ 0215

GO

4-642#

N 6

SEQ 0217

OM	4-656#	10-151	10-151	10-151	10-173	10-173	10-173	10-223	10-223	10-241	10-241	10-268	10-268	10-281
	10-281	10-327	10-327	10-327	10-350	10-350	10-350	10-407	10-407	10-431	10-431	10-483	10-483	10-483
	10-506	10-506	10-506	10-531	10-531	10-531	10-547	10-547	10-547	10-583	10-583	10-583	10-603	10-603
	10-603	10-675	10-675	10-675	10-675	10-705	10-705	10-756	10-756	10-779	10-779	10-837	10-837	10-837
	10-859	10-859	10-859	10-914	10-914	10-914	10-937	10-937	10-937	10-989	10-989	10-:53	10-:53	10-:03
	10-:03	10-:20	10-:20	10-:69	10-:69	10-:89	10-:89	10-<84	10-<84	10-<84	10-<84	10-=09	10-=09	10-=09
	10-=09	10-=81	10-=81	10-=99	10-=99	10->53	10->53	10->73	10->73	10-?41	10-?41	10-?63	10-?63	10-a30
	10-a30	10-a60	10-a60											
OPE	4-739#													
OPI	4-683#													
OPPR	7-0#	10-151*	10-173*	10-223*	10-241*	10-327*	10-350*	10-407*	10-431*	10-483*	10-506*	10-531*	10-547*	10-583*
	10-603*	10-675*	10-675*	10-756*	10-779*	10-837*	10-859*	10-914*	10-937*	10-<84*	10-=09*	10-=81*	10-=99*	10->53*
	10->73*	10-?41*	10-?63*	10-a30*	10-a60*	25-35	25-40	25-45						
OR	4-625#													
PAR	4-673#													
PAT	4-622#													
PFECH	15-1	15-1#												
PFECH1	15-1	15-1#												
PFECH2	15-1	15-1#												
PFECH3	15-1	15-1#												
PFECH4	15-1	15-1#												
PFTSTN	15-1	15-1#	15-1*											
PGE	4-628#													
PGM	4-660#	10-48	10-48	10-48	10-48	10-48	10-48	10-151	10-151	10-173	10-173	10-223	10-241	10-268
	10-281	10-327	10-327	10-327	10-350	10-350	10-350	10-407	10-431	10-483	10-483	10-483	10-483	10-506
	10-506	10-506	10-506	10-531	10-531	10-547	10-547	10-583	10-583	10-603	10-603	10-675	10-675	10-675
	10-675	10-675	10-675	10-705	10-756	10-756	10-779	10-779	10-837	10-837	10-859	10-859	10-914	10-914
	10-937	10-937	10-:03	10-:20	10-:69	10-:89	10-<84	10-<84	10-<84	10-=09	10-=09	10-=09	10->53	10->73
	10-?41	10-?63	10-a30	10-a30	10-a60	10-a60								
PIP	4-664#	10-151	10-151	10-151	10-173	10-173	10-173	10-223	10-223	10-241	10-241	10-268	10-268	10-281
	10-281	10-327	10-327	10-327	10-350	10-350	10-350	10-407	10-407	10-431	10-431	10-483	10-483	10-483
	10-506	10-506	10-506	10-531	10-531	10-531	10-547	10-547	10-547	10-583	10-583	10-583	10-603	10-603
	10-603	10-675	10-675	10-675	10-675	10-705	10-705	10-756	10-756	10-779	10-779	10-837	10-837	10-837
	10-859	10-859	10-859	10-914	10-914	10-914	10-937	10-937	10-937	10-989	10-989	10-:53	10-:53	10-:03
	10-:03	10-:20	10-:20	10-:69	10-:69	10-:89	10-:89	10-<84	10-<84	10-<84	10-<84	10-=09	10-=09	10-=09
	10-=09	10-=81	10-=81	10-=99	10-=99	10->53	10->53	10->73	10->73	10-?41	10-?41	10-?63	10-?63	10-a30
	10-a30	10-a60	10-a60											
PIRQ	4-601#													
PIRQVE	4-601#													
PORTA	7-0#	9-35*	9-36	9-40	9-43	9-47	9-51	9-54	10-33	10-33	10-40	10-40	10-48	10-48
	10-52	10-151	10-151	10-151	10-151	10-151	10-151	10-151	10-151	10-173	10-173	10-173	10-173	10-173
	10-173	10-173	10-173	10-173	10-173	10-223	10-223	10-223	10-223	10-223	10-223	10-223	10-223	10-223
	10-223	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-268	10-268	10-268	10-268
	10-268	10-268	10-268	10-281	10-281	10-281	10-281	10-327	10-327	10-327	10-327	10-327	10-327	10-327
	10-327	10-327	10-327	10-327	10-327	10-327	10-327	10-327	10-327	10-350	10-350	10-350	10-350	10-350
	10-350	10-350	10-350	10-350	10-350	10-350	10-350	10-407	10-407	10-407	10-407	10-407	10-407	10-407
	10-407	10-407	10-407	10-407	10-407	10-407	10-407	10-431	10-431	10-431	10-431	10-431	10-431	10-431
	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483	10-483
	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-506	10-531	10-531
	10-531	10-531	10-531	10-531	10-531	10-531	10-531	10-531	10-531	10-547	10-547	10-547	10-547	10-547
	10-547	10-547	10-583	10-583	10-583	10-583	10-583	10-583	10-583	10-583	10-583	10-583	10-603	10-603
	10-603	10-603	10-603	10-603	10-603	10-603	10-603	10-603	10-603	10-675	10-675	10-675	10-675	10-675
	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-675
	10-675	10-675	10-675	10-675	10-675	10-675	10-675	10-705	10-705	10-705	10-705	10-705	10-756	10-756
	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-756	10-779	10-779
	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-779	10-837	10-837	10-837

10-837 10-837 10-837 10-837 10-837 10-837 10-837^L 6 10-837 10-837 10-837 10-837 10-859 10-859 10-859
SEQ 0221

PR6

4-601# 9-27

N 6

SEO 0223

10-407* 10-407* 10-407* 10-407* 10-407* 10-431 10-431^C 7 10-431* 10-431* 10-431* 10-431* 10-431* 10-431* 10-431* 10-431*
SEG 0225

10-241 10-241 10-241 10-241 10-241* 10-268 10-268^E 7 10-268 10-281 10-281 10-281 10-327 10-327 10-327
SEO 0227

SCOPE 4-601# 10-62 10-151 10-173 10-223 10-241 10-268^G 10-281⁷ 10-327 10-350 10-407 10-431 10-483 10-506
SEQ 022⁶

TEST 17 10-545 10-545 10-545#

1 7

SEQ 0231

TST17 10-545# 26-8

K 7

SFO 0233

TST1AA	10-17#	11-1												
TST2	10-82#	26-8												
TST20	10-565#	26-8												
TST21	10-601#	26-8												
TST22	10-620#	26-8												
TST23	10-689#	26-8												
TST24	10-726#	26-8												
TST25	10-777#	26-11												
TST26	10-803#	26-11												
TST27	10-857#	26-11												
TST3	10-171#	26-8												
TST30	10-880#	26-11												
TST31	10-935#	26-11												
TST32	10-953#	26-11												
TST33	10-:08#	26-11												
TST34	10-:68#	26-11												
TST35	10-:18#	26-11												
TST36	10-:39#	26-11												
TST37	10-:87#	26-14												
TST4	10-189#	26-8												
TST40	10-<41#	26-14												
TST41	10-=07#	26-14												
TST42	10-=29#	26-14												
TST43	10-=97#	26-14												
TST44	10->17#	26-14												
TST45	10->71#	26-14												
TST46	10->96#	26-14												
TST47	10-?61#													
TST5	10-239#	26-8												
TST50	10-?90#													
TST50B	10-a30	10-a30	10-a31#											
TST51	10-a31	10-a58#												
TST51B	10-a60	10-a60	10-a61#											
TST52	10-a61	10-a65#												
TST6	10-256#	26-8												
TST7	10-279#	26-8												
TSTADR	9-91	26-5#												
TSTERR	23-10#													
TSTNUM	7-0#	14-1*	25-32	25-33	25-34	25-35	25-36	25-37	25-38	25-39	25-40	25-41	25-42	25-43
	25-44	25-45	25-46	25-48										
TYPDS	11-1	11-1	15-1	22-1#										
TYPE	9-6	9-29	9-33	9-38	9-46	9-48	9-50	9-61	9-69	9-80	9-87	9-102	9-105	9-106
	11-1	11-1	11-1	14-1	14-1	15-1	15-1	15-1	15-1	15-1	15-1	15-1	16-1	17-1
	18-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1	19-1
	19-1	19-1	19-1	19-1	19-1	20-1	20-1	22-1#						
TYPOC	9-8	9-104	15-1	15-1	19-1	22-1#								
TYPON	22-1#													
TYPOS	9-47	9-49	22-1#											
U0	4-618#													
U1	4-619#													
U3	4-620#													
UNS	4-684#													
UPE	4-631#													
VV	4-657#	10-151	10-151	10-151	10-151	10-173	10-173	10-173	10-173	10-223	10-223	10-223	10-223	10-223
	10-223	10-223	10-241	10-241	10-241	10-241	10-241	10-241	10-241	10-268	10-268	10-268	10-281	10-281
	10-281	10-327	10-327	10-327	10-327	10-327	10-327	10-350	10-350	10-350	10-350	10-407	10-407	10-407

10-431 10-431 10-431 10-483 10-483 10-483 10-483^{M 7} 10-483 10-483 10-506 10-506 10-506 10-506 10-506
SEQ 0235

RELEAS 4-203# 10-223 10-241 10-327 10-327 10-350 10-350^D 10-407⁸ 10-431 10-483 10-506 10-531 10-547 10-583
SEQ 0239

